

Trabajo Fin de Grado

Grado en Ingeniería Aeroespacial

Estudio de Mejora en el Posicionamiento de Objetos para Sistemas de Navegación Aérea

Autor: Marta Cobo Quirós

Tutor: María Ángeles Martín Prats

Dep. de Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2016



Trabajo Fin de Grado
Grado en Ingeniería Aeroespacial

Estudio de Mejora en el Posicionamiento de Objetos para Sistemas de Navegación Aérea

Autor:

Marta Cobo Quirós

Tutor:

María Ángeles Martín Prats

Profesor Titular

Dep. de Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2016

Trabajo Fin de Grado: Estudio de Mejora en el Posicionamiento de Objetos para Sistemas de Navegación Aérea

Autor: Marta Cobo Quirós
Tutor: María Ángeles Martín Prats

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

Hay que perseverar, y sobre todo, tener confianza en uno mismo.

MARIE CURIE

Quizá este sea el capítulo que más nos cueste a todos escribir. Incluso mucha gente opta por obviarlo, porque no es fácil agradecer a tanta gente que se ha volcado contigo en el duro recorrido como ingeniero.

En primer lugar, me gustaría agradecer a mi familia, por tanta paciencia, tanto esfuerzo y tanto orgullo como sé que sienten por mí. A mis padres, por tanto apoyo. A mi hermana, por ser mi cómplice.

A mis amigos por tantas risas, lágrimas, apuntes, madrugones y noches sin dormir. Y por todas esas noches en las que cualquier excusa era buena para celebrarlo. Por tantos "Una más y nos vamos".

A ti, por hacerme valiente. Por acompañarme, apoyarme y ayudarme siempre. Por hacerme ver que sí se puede.

Agradecer también a mi tutora, María Ángeles Martín Prats, por haberme dado esta oportunidad, y haber confiado en mí.

Y por supuesto, a Ignazio Federico Finazzi, pues sin su ayuda no habría sido posible.

Marta Cobo Quirós

Sevilla, 2016

Resumen

En los últimos años la ocupación del espacio aéreo ha aumentado notablemente, y se prevee que este aumento siga creciendo. Del mismo modo, y prácticamente en paralelo, ha crecido a nivel europeo la preocupación de la congestión de nuestro espacio aéreo. Desde el plan SESAR y NextGen, son múltiples las mejoras en navegación que pretenden desarrollarse, además de la unificación de los sistemas para hacerlos compatibles. Una de las mejoras que se proponen se centra en el sistema anticolidión de las aeronaves. Su objetivo es el desarrollo de un nuevo sistema, el ACAS X, cuyo nombre es aún temporal. Su variable ACAS X_U está orientada a lograr detectar y evitar objetos en tiempo real y a dotar de autonomía a los sistemas.

Puesto que esta preocupación nos concierne a todos, desde este Trabajo Fin de Grado, se pretende estudiar una posible mejora del posicionamiento de objetos en los sistemas de navegación. La propuesta consiste en ofrecer una información adicional a los sistemas, de forma que se logre hacerlos más redundantes. Para ello, a través del procesamiento digital de imágenes, se obtendrá la información necesaria del posicionamiento de los objetos. Mediante la herramienta *Matlab* se han desarrollado una serie de algoritmos que hacen posible el cálculo de la distancia a los objetos detectados en imágenes.

A largo plazo, se trata de una propuesta de carácter ambicioso. El presente trabajo sólo constituye los primeros pasos en el largo recorrido que se plantea.

Índice

<i>Resumen</i>	III
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	2
1.3 Estructura del documento	2
2 El sistema ACAS	3
2.1 Reseña Histórica	3
2.2 ACAS y TCAS	5
2.3 Descripción del equipo	5
2.3.1 Principios de Funcionamiento	5
2.3.2 Equipo físico	5
2.4 Tipos de ACAS	7
2.4.1 ACAS I	7
2.4.2 ACAS II	8
TCAS II 6.02 y 6.04A	9
TCAS II 7.0	9
TCAS II 7.1	9
2.4.3 ACAS III	12
2.4.4 ACAS X	12
2.4.5 Sistemas anticolidión en UAV	13
3 Procesamiento digital de imágenes	15
3.1 Espacios de color	15
3.2 Detección de objetos	17
3.2.1 Descriptores de imagen	17
3.2.2 Clasificación de los descriptores	17
3.2.3 Descriptores de color	18
Dominant Color Descriptor	18
Color Structure Descriptor	20
Color Layout Descriptor	20
3.2.4 Descriptores de textura	21
3.2.5 Descriptores de forma	22
3.3 Detección de borde	22
3.3.1 Algoritmos de detección de bordes en imagenes monocromáticas	23
4 Desarrollo del proyecto y resultados	27
4.1 Detección de borde	27
4.2 Detección de objetos	34

4.3	Identificación de objetos	38
4.4	Cálculo de distancia	40
4.5	Resultados	44
5	Conclusiones	49
6	Líneas futuras	51
	<i>Índice de Figuras</i>	53
	<i>Índice de Tablas</i>	55
	<i>Bibliografía</i>	57
	<i>Glosario</i>	59

1 Introducción

Nunca vayas por el camino trazado, porque conduce hacia donde otros han ido ya.

ALEXANDRE GRAHAM BELL

Este primer capítulo recoge las razones que han llevado a la realización de este Trabajo Fin de Grado, cuáles son los objetivos que se plantean y qué estructura presenta el documento.

1.1 Motivación

«Nuestras compañías aéreas y sus clientes han tenido que sufrir más de diez años de servicios reducidos y plazos no respetados en la senda hacia el cielo único europeo. [...]», declaró en 2013 Siim Kallas, el por entonces Comisario Europeo de Transportes. El Cielo Único Europeo, (Single European Sky (SES)) es una ambiciosa iniciativa de la Comisión Europea en la búsqueda de una reorganización del espacio aéreo europeo. La saturación de la capacidad del espacio aéreo es un hecho real: existe una predicción de aumento del número de vuelos de un 50 % en los próximos 10 a 20 años. La fragmentación del espacio aéreo europeo suponen unas pérdidas de 5000 millones de euros anuales a compañías y clientes. Mediante el SES se pretende reformar el sistema de control del tráfico aéreo en Europa, cumpliendo cuatro pilares básicos. Uno de estos pilares es la mejora de la seguridad.[18]

La seguridad sigue siendo la primera prioridad en la aviación. El aumento de la seguridad o la mejora de los sistemas que promueven la seguridad de la aeronave, está estrechamente relacionado con una mejora en la capacidad. En concreto, el sistema anticolidión de las aeronaves tiene un papel fundamental en este aspecto. Por un lado, una mejora en el sistema podría conllevar una reducción en las distancias mínimas de seguridad, con el consiguiente aumento de la capacidad en el espacio. Por otro lado, el aumento de vehículos aéreos no tripulados hace necesaria la expansión de estos sistemas a dichos vehículos. Enmarcado en estos aspectos, se plantea la primera de las razones que han motivado la realización de este proyecto.

Proponer un sistema que sustituya al anterior resultaría inviable, debido al alto coste que supone una innovación de tal envergadura, sin contar con la gran inversión que ya se realizó para equipar las aeronaves con el sistema actual. Por ello, para aquellas aeronaves que ya dispongan de un sistema embarcado encargado de mantener la seguridad y evitar una posible colisión, se les puede dotar de una redundancia adicional. Además, debido al auge en el mercado de los Unmanned Aerial Vehicles (UAVs), es necesario un equipo que no sólo detecte aeronaves cooperativas, si no que detecte también cualquier objeto que se encuentre en el espacio aéreo.

El presente proyecto se encuentra también motivado por el grupo de investigación del Departamento de Ingeniería Electrónica de la Escuela Técnica Superior de Ingeniería. Tras más de 10 años de trabajo, disponen de una plataforma de navegación que complementa las señales con información obtenida a partir del procesamiento digital de imágenes. [7] Enfocado a ello se propone mejorar el sistema anticolidión actual, Airborne Collision Avoidance System (ACAS), con una entrada de información adicional. El objetivo es aprovechar el avance conseguido en computación y emplear la visión computacional para extraer información de las imágenes. Concretamente, se pretende conseguir la detección de objetos en imágenes, y a partir de ello, se podría obtener su posición, forma, velocidad, etc.

1.2 Objetivos

El objetivo general es el estudio de una posible mejora de los sistemas de navegación aérea. Con el fin de dotar de una redundancia adicional a dichos sistemas, se busca lograr un flujo de información adicional que pueda hibridarse con las señales de las que ya se dispone. Se desea también que sea aplicable a cualquier tamaño de aeronave, incluyendo además aquellos dispositivos que no tengan previamente integrado determinados equipos de navegación. Por ejemplo, para aquellos que ya dispongan de sistema TCAS, se podrá complementar las características que tiene actualmente, y así mejorar el posicionamiento de objetos. Para otro tipo de vehículos, como los no tripulados, se trataría de un dato de entrada fundamental para dotarlos de autonomía y así lograr un cambio de trayectoria en tiempo real.

La aportación de este Trabajo Fin de Grado consiste en comprobar si ese flujo de información adicional se podría obtener a partir del procesamiento digital de imágenes y la visión computacional. Dentro de esta amplia área, el objetivo principal del proyecto se centra en la detección de objetos fijos en imágenes y el cálculo de distancia a ellos.

1.3 Estructura del documento

El documento está constituido por seis capítulos, incluyendo el capítulo de introducción. El segundo capítulo recoge el estado del arte en el que se sitúa el proyecto. En él se hace un barrido por los sistemas anticolidión, partiendo de las primeras colisiones que motivaron su creación y los primeros equipos que se desarrollaron, hasta llegar al sistema actual y las líneas futuras. En el tercer capítulo se desarrolla un estudio del procesamiento digital de imágenes, que permite familiarizarse con este campo. En el cuarto capítulo se plantean todos los pasos seguidos hasta el objetivo último del cálculo de distancia a objetos y se analizan los resultados. Por último, el documento se cierra con las conclusiones generales obtenidas y las posibles mejoras y líneas futuras que ofrece.

2 El sistema ACAS

En este capítulo se presenta el sistema ACAS, cuál es la descripción del equipo y qué tipos hay.

ACAS es el sistema de aviónica embarcado encargado de evitar colisiones. Está basado en señales de transpondedor de radar secundario, (Secondary Surveillance Radar (SSR)), de manera que interroga a las aeronaves detectadas para recibir información de su posición, altitud, etc. Actúa de forma independiente al Air Traffic Control (ATC) y se emplea como último recurso de seguridad en la reducción del riesgo de colisión.

2.1 Reseña Histórica

Con el paso del tiempo, el tráfico aéreo se ha ido incrementado continuamente. Se espera, además, que esta tendencia continúe en los próximos años. Motivado por el gran desarrollo en los sistemas de control del tráfico aéreo, se ha podido hacer frente a este aumento de tráfico sin necesidad de reducir los niveles de seguridad.

A pesar de los avances en los sistemas ATC, el riesgo de que se produzca un accidente, ya sea por un error humano o técnico, sigue existiendo. De hecho, en los años cincuenta, se produjeron algunas colisiones entre aeronaves por pérdida de separación. Estos hechos fueron determinantes para el desarrollo del concepto de un sistema anticolidión embarcado.

El 30 de junio de 1956, tuvo lugar un accidente entre dos aviones de pasajeros, un Douglas DC-7 y un Lockheed L-1049. Cuando sobrevolaban el Gran Cañón del Colorado, las adversas condiciones meteorológicas les hicieron colisionar. Ambos pilotos iniciaron maniobras para esquivarlas sin tener conciencia de que había otra aeronave en su misma situación; cuando lograron avistarse entre ellos, era demasiado tarde para evitar la colisión. Este hecho movilizó tanto a las aerolíneas como a las autoridades aeronáuticas, quienes promovieron el desarrollo de un sistema anticolidión. Ese mismo año, John S. Morrel ¹propuso emplear el concepto en el que hoy día está basado el sistema ACAS.

Durante la década de los sesenta y principios de los setenta, varios fabricantes desarrollaron un prototipo del sistema anticolidión para aeronaves. Sin embargo, aunque durante los ensayos funcionaron correctamente, se concluyó que no era adecuado para operaciones habituales de aerolíneas. En la decisión se tuvo en cuenta que podría generar un elevado número de alertas innecesarias en áreas con tráfico denso.

A mediados de los setenta, y aprovechando la tecnología existente, se desarrolló el Beacon Collision

¹ John "Smiley" Morrel fue un ingeniero de Bendix, empresa líder en el establecimiento de estándares de seguridad en sistemas de control y carga de frenos. A mediados de los años 50 desarrolló un concepto, al que llamó τ , que medía el riesgo de colisión en función del número de segundos al que se encontraba el intruso más cercano. Este concepto fue clave en el desarrollo de los sistemas anticolidión, hasta tal punto que hoy día se sigue teniendo en cuenta.

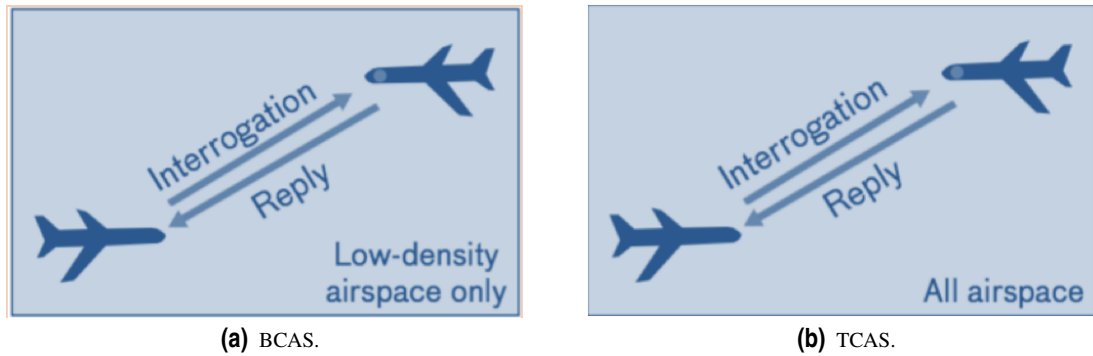


Figura 2.1 En la imagen a) el sistema BCAS (1970); en la b), el TCAS (1980-2000). *Fuente: [13].*

Avoidance System (BCAS), el predecesor del Traffic Alert and Collision Avoidance System (TCAS). Este sistema disponía de información relativa a la posición, velocidad y ruta de las aeronaves intrusas, pero sólo en una región del espacio aéreo con baja densidad de aeronaves. Para ello, se basaba en el sistema de tierra Air Traffic Control Radar Beacon System (ATCRBS). Éste interrogaba a las aeronaves próximas y, con la respuesta de sus transpondedores, obtenía todos los datos que proporcionaba al BCAS.

El 25 de septiembre de 1978, se produjo otra colisión entre dos aeronaves en aire. La colisión se produjo entre un Boeing 727 y un Cessna 172 en San Diego, California. La imagen 2.2 muestra la caída de uno de los aviones tras la colisión. De nuevo, una colisión incentivó a la Federal Aviation Administration (FAA) a iniciar el desarrollo definitivo del sistema TCAS. En 1981, surgió dicho sistema, que mejoraba el diseño básico del BCAS y proporcionaba nuevas capacidades, sobre todo en espacios aéreos con elevado número de aeronaves. En la imagen 2.1 se muestra la principal diferencia entre ambos sistemas, el alcance que tenían en cuanto a densidad de tráfico.

En 1986, tuvo lugar una nueva colisión entre un Douglas DC-9 y un Piper PA-28 Archer en Cerritos, California. Esta vez, el accidente determinó la obligatoriedad del uso del TCAS. El Congreso de los Estados Unidos y la FAA acordaron hacerlo obligatorio para ciertas aeronaves de EEUU y aeronaves extranjeras que operasen en espacio aéreo americano.

En paralelo al desarrollo del TCAS, la International Civil Aviation Organization (ICAO) fue desarrollando los estándares del sistema ACAS. Finalmente, el 11 de noviembre de 1993, reconoció oficialmente dicho sistema, incluyendo su descripción en el Anexo 2 [10] y la regulación de su uso en el Anexo 6 [21]. Desde 1995, incluye también sus estándares en el Anexo 10 [8].



Figura 2.2 Accidente aéreo en San Diego, en 1978. *Fuente: [13].*

A partir de entonces, FAA, ICAO y otras organizaciones de autoridad aeronáutica como EUROCONTROL (European Organization for the Safety of Air Navigation), comenzaron a imponer el uso de este sistema en todo tipo de aeronaves. Además, iniciaron un desarrollo continuo para mejorar dicho sistema.

2.2 ACAS y TCAS

Antes de continuar profundizando en el sistema ACAS, se plantea necesaria la aclaración de los términos TCAS y ACAS. Ambos son mencionados a lo largo de este capítulo, y aunque parezcan ser lo mismo, lo cierto es que no lo son.

El término ACAS hace referencia al sistema en si mismo, al estándar reconocido por ICAO y al concepto de anticollisión; TCAS es la implementación física de dicho concepto, el equipo embarcado en la aeronave.

2.3 Descripción del equipo

2.3.1 Principios de Funcionamiento

ACAS está diseñado para trabajar con autonomía e independencia de los equipos de navegación aérea; tampoco necesita ningún sistema de tierra, aunque sí interaccione con ellos. Gracias a los avisos que el sistema emite, los pilotos conocen la situación del tráfico que tienen alrededor. Esto les ayuda a la hora de avistar aeronaves cercanas y evitar la colisión contra ellas.

A través de antenas mediante Radio Frequency (RF), el sistema interroga al transpondedor de la aeronave que se encuentre próxima a ella. En base a las señales que recibe, el sistema es capaz de determinar su altitud y posición, además de la cantidad de tráfico que le rodea.

ACAS emite dos tipos de alertas: avisos de tránsito y avisos de resolución.

- **Traffic Advisories (TA)** están destinados a ayudar al piloto en el conocimiento de las aeronaves en conflicto, y le preparan para posibles avisos de resolución.
- **Resolution Advisories (RA)** se emite cuando existe riesgo de colisión. Le recomiendan al piloto las maniobras a realizar para evitar el conflicto. Un RA le dirá al piloto la altitud a la que debe volar para evitar la aeronave intrusa. En el caso de que dicha aeronave esté equipada con el mismo sistema, ambas se coordinaran poniendo el transpondedor en enlace de datos Modo S², para seleccionar resoluciones complementarias al conflicto.

La emisión de uno de estos dos avisos dependerá de los datos de posición y velocidad que reciba tras interrogar a las aeronaves vecinas. Emitirá un TA si se va a acercar lo suficiente, y emitirá un RA si el acercamiento será tal que traspase los límites de seguridad establecidos. Dicha identificación se verá influenciada por la altitud, de forma que tendrá varios niveles de sensibilidad. Por ejemplo, no se pueden comparar situaciones de vuelo de crucero con aproximaciones, donde el tráfico aéreo será mucho mayor.

2.3.2 Equipo físico

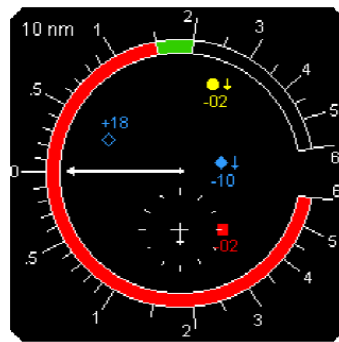
El equipo físico dispone de una interfaz que facilita al piloto la visualización de las aeronaves intrusas. Puede ser de tres formas distintas [11]:

² Modo S o selectivo permite establecer un enlace de datos con cada aeronave de forma selectiva.

- Un instrumento dedicado en exclusiva. (2.3a)
- Un instrumento combinado con el indicador de velocidad vertical.(2.3b)
- Un instrumento combinado con la pantalla de navegación o el indicador de posición horizontal. (2.3c)



(a) Instrumento dedicado en exclusiva.



(b) Instrumento combinado con el indicador de velocidad vertical.



(c) Instrumento combinado con el indicador de posición horizontal.

Figura 2.3 Tipos de interfaz para TCAS. Fuente: [11].

En la imagen 2.3 se pueden observar las tres interfaces mencionadas anteriormente.

En cuanto a la simbología empleada en la interfaz, el mapa mostrado se completa con todas las aeronaves intrusas, según su altitud, la distancia a ellas y su trayectoria (si es ascendente o descendente).

El tráfico se representa con pequeñas figuras geométricas asociadas a ciertos números. Estos indican la separación respecto a la aeronave, expresada en cientos de ft. Por ejemplo, la figura 2.4a representa la propia aeronave; la figura 2.4b representa una aeronave intrusa de altitud desconocida, y, por ello, no tiene ningún número asociado; la figura 2.4c muestra un intruso próximo a 200 ft por debajo y descendiendo; en la figura 2.4d se representa el símbolo de un aviso de tráfico TA a 700ft por encima y en la figura 2.4e el símbolo de un aviso de resolución RA a 100ft por debajo y ascendiendo.

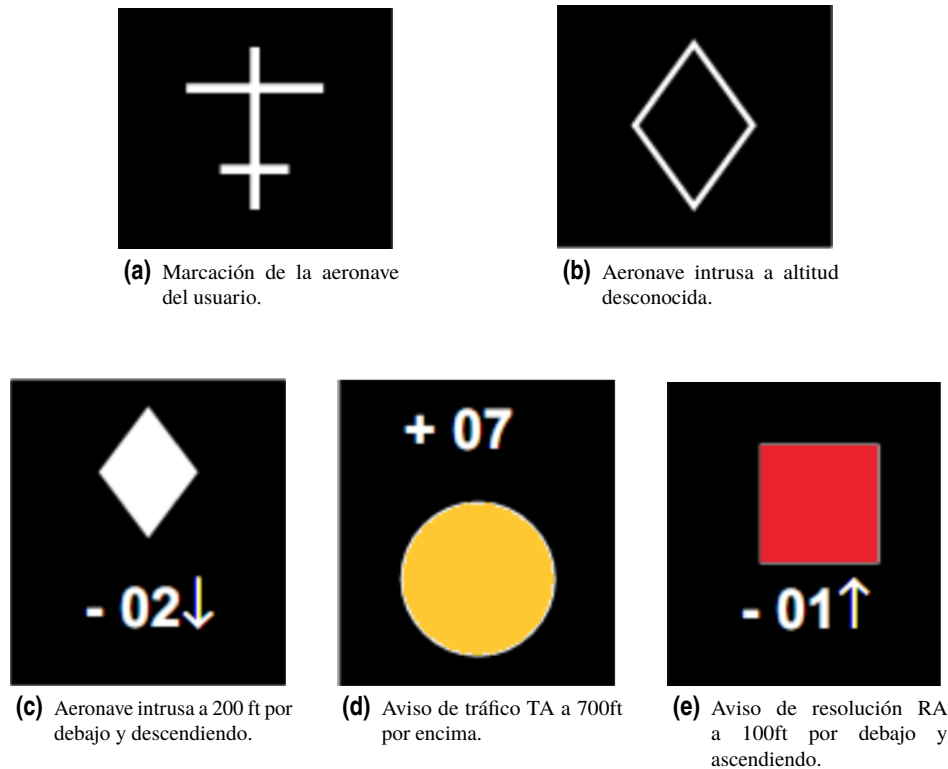


Figura 2.4 Simbología empleada por TCAS. *Fuente: [11].*

2.4 Tipos de ACAS

Según el Volumen IV del Anexo 10 de ICAO [8], actualmente existen tres tipos de ACAS distintos:

- **ACAS I** proporciona ayuda al piloto en el avistamiento de aeronaves, pero no tiene la capacidad de emitir RAs.
- **ACAS II** proporciona TA y RA en el plano vertical.
- **ACAS III** proporciona TA y RA tanto en el plano vertical como en el horizontal.

Aunque el ACAS III sea considerado en el Anexo 10 de ICAO [8] el sistema futuro, lo cierto es que no se llegará a desarrollar. Se debe a las dificultades técnicas que planteaba la determinación horizontal de las aeronaves con suficiente precisión. En contraposición, el futuro real de este sistema anticolidión se encuentra en el ACAS X, basado en tecnología Automatic Dependent Surveillance-Broadcast (ADS-B), del cual se hablará más adelante.

2.4.1 ACAS I

ACAS I es un sistema anticolidión que proporciona ayudas en la visualización de aeronaves que se encuentran próximas. Sin embargo, no es capaz de proporcionar ningún aviso específico anticolidión. La maniobra para evitar la colisión se deja a criterio de los pilotos, por lo que las operaciones de aquellas aeronaves equipadas con ACAS I no pueden coordinarse con las de su sucesor, el ACAS II.

Tiene tres niveles de alertas:

- TA
- Proximity Advisories (PA)
- Otros avisos de tráfico en general, sin necesidad de que puedan constituir un peligro.

El criterio seguido para la emisión de uno de estos avisos se basa en el correcto avistamiento de las aeronaves vecinas por parte de la tripulación. Su objetivo es proporcionar dicha visión con la suficiente antelación para poder maniobrar y evitar la colisión, mediante datos de altitud y velocidad vertical.

TAs son emitidos según la proximidad a la aeronave intrusa, usando dos niveles de sensibilidad. Todos los intrusos que estén equipados con transpondedor y se encuentren a cinco millas náuticas (NM) se mostrarán en la pantalla de tráfico. Además, estos avisos van acompañados de una alerta sonora "*Traffic, Traffic*" para informar a la tripulación de que un TA ha sido emitido. Estas alertas serán inhibidas si el tren de aterrizaje está extendido o si una aeronave equipada con radioaltímetro se encuentra a 400 ft sobre el nivel del suelo.

En cuanto al equipo embarcado, la única implementación física desarrollada fue el TCAS I. En Europa nunca ha sido obligatoria su instalación, y no tiene reglas operacionales establecidas. En Estados Unidos sí es obligatoria, pero solo para aeronaves pequeñas.

2.4.2 ACAS II

ACAS II es un sistema anticolidión basado en señales de transpondedor de SSR, que interroga en Modo C³ y Modo S a los transpondedores de las aeronaves cercanas. Proporciona TA y RA en el plano vertical, de forma que indica la velocidad vertical que el piloto debería alcanzar para que la colisión no tuviese lugar. Es obligatorio su uso en la mayor parte del mundo; en Europa, por ejemplo, para aeronaves con más de 19 asientos o con un Maximum Take-Off Weight (MTOW) superior a 5700 kg. [23]

El sistema ACAS II determina la posición de las aeronaves vecinas, entendiéndose por aeronave vecina aquella que se encuentra a menos de 14 NM⁴ y a menos de 10000 ft en vertical, arriba o abajo. Con estas medidas delimita el volumen de protección en el que se encuentra sumergida la aeronave, como puede verse en las imágenes 2.5 y 2.6. Gracias a la respuesta que obtiene de la interrogación a sus transpondedores, a través de la altitud presión obtenida, puede estimar la posición horizontal de éstas. Con sucesivas estimaciones espaciadas en el tiempo, la aeronave es capaz de predecir su posición futura.

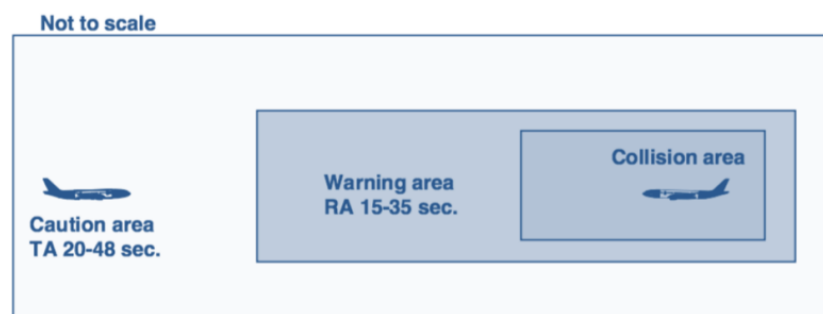


Figura 2.5 Volumen de protección en vista vertical. *Fuente:* [11].

³ El Modo C interroga sobre la altura barométrica de la aeronave.

⁴ Posibilidad de reducirse a 5 NM en situaciones con tráfico denso

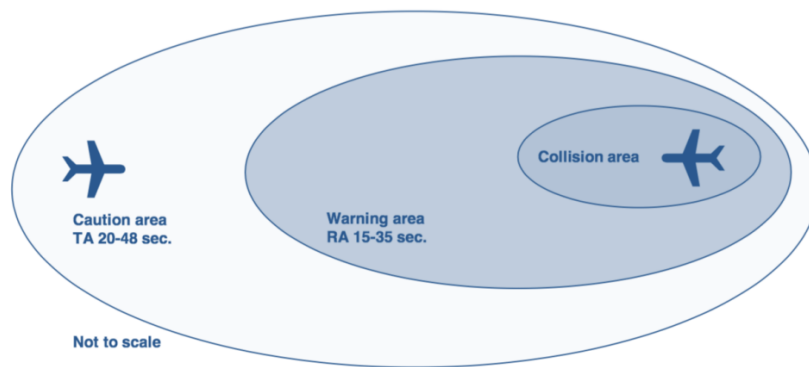


Figura 2.6 Volumen de protección en vista horizontal. *Fuente: [11].*

Cuando el sistema ha determinado cuáles son las aeronaves vecinas y dónde se encuentran, tomará la decisión de si emitir un aviso y de cuál emitir. Si alguna aeronave se va a aproximar lo suficiente, emitirá un TA; si dicha aproximación se puede llegar a considerar un peligro, emitirá un RA, a no ser que haya incertidumbre en la predicción de la trayectoria. Para ello, el ACAS II cuenta con siete niveles de sensibilidad aplicables en diferentes rangos de altitudes. Por ejemplo, por encima de 41000 ft se considera que una aeronave representa un peligro si se encuentra a una distancia horizontal menor de 1.1 NM, vertical menor de 800 ft y la aproximación más cercana se produciría en 35 segundos; por debajo de 2300 ft, estas restricciones aumentan a 0.2 NM, 600 ft y 15 s.

TCAS II 7.0 y TCAS II 7.1 son las dos implementaciones que actualmente se encuentran en funcionamiento, pero no son las únicas versiones que surgieron de ACAS II. [2]

TCAS II 6.02 y 6.04A

En septiembre de 1989 se completó el diseño de la versión 6.02 y se puso en funcionamiento en abril del año siguiente.

ICAO organizó una evaluación operacional del sistema a nivel mundial con objeto de determinar el funcionamiento del TCAS II a finales de los años ochenta. Como resultado de esta evaluación, surgieron una gran cantidad de mejoras que llevaron al desarrollo de la nueva versión 6.04A en 1993. Con esta nueva versión se reducía el número de avisos molestos, que se producían a baja altitud o en situaciones de bajada de nivel de vuelo.

TCAS II 7.0

Tras la versión 6.04A, en diciembre de 1997 se aprobó la nueva versión 7.0, haciéndose operativa a principios de 1999.

El objetivo de esta nueva versión era promover la compatibilidad con los demás sistemas de control del tránsito aéreo. Algunas de las mejoras más significativas que incluyó esta nueva versión son, por ejemplo, la reducción de las interferencias electromagnéticas y la compatibilidad con el sistema RVSM. También simplifica las alertas sonoras para facilitar su comprensión y permite la revocación de un RA en encuentros que estén coordinados.

TCAS II 7.1

En esta versión se incluyen dos cambios significativos para mejorar el TCAS II. Por un lado se incluye una nueva alerta sonora RA para evitar ciertas confusiones, “*Level Off*”; por otro, las aeronaves equipadas con

esta versión podrán corregir su actuación cuando siga existiendo riesgo de colisión por las maniobras de la otra aeronave.

En la versión anterior, la alerta sonora *"Adjust vertical speed, adjust"* provocaba en ciertos casos que algunos pilotos no respondieran correctamente. La respuesta errónea provocaba que el riesgo de colisión aumentase debido a que en lugar de disminuir el nivel de vuelo lo aumentaban, y viceversa. El principal motivo era la ambigüedad que ofrecía esta alerta para situaciones que necesitaban maniobras concretas. Este hecho, sumado a que estas alertas suponían dos tercios del total de RA, aumentaba notablemente el número de incidentes con riesgo de colisión. En la imagen 2.7, se observan dos situaciones de peligro. En la figura 2.7a, la aeronave que recibe la alerta de ajuste de velocidad vertical, en lugar de disminuirla a 1000 ft/min la aumenta a 4500ft/min, originando una separación de sólo 350 ft y por lo tanto, una situación de peligro.

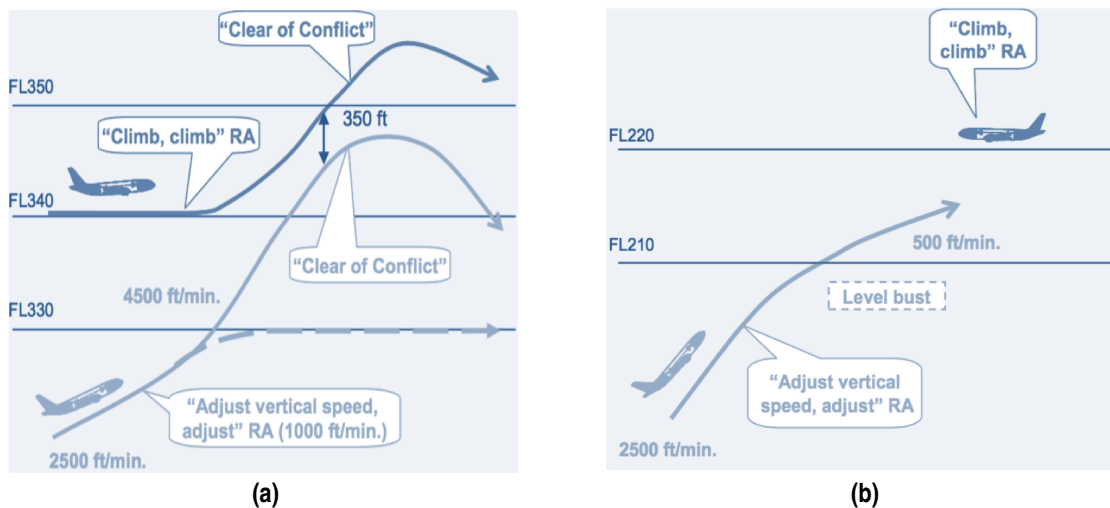


Figura 2.7 Situaciones de peligro generadas por la malinterpretación del aviso *"Adjust Vertical Speed"*.
Fuente: [2].

En otras ocasiones, lo que ocurría es que ciertas aeronaves, siguiendo el aviso RA, se cruzaban de nivel de vuelo. Esto provocaba que a otras aeronaves les apareciese la misma alerta, lo que perturbaba las operaciones del ATC. Esta situación se muestra en la imagen 2.7b.

Para solucionar estos problemas, en la nueva versión 7.1 se incluyó una nueva alerta sonora que sustituiría a la antigua. La alerta es *"Level Off, Level Off"*, que se trata de una alerta intuitiva y regulada que indica al piloto qué debe hacer. Además, incluye una modificación en cuanto a su significado que se muestra en la imagen 2.8. Con el anterior aviso, se podía variar la velocidad vertical entre un rango de 0, 500, 1000 y 2000 ft/min. En la nueva versión, el RA significa que la velocidad vertical sea de 0 ft/min. Este cambio también minimiza las desviaciones en la altitud de las aeronaves, lo que supone un alivio de trabajo para las operaciones del ATC.

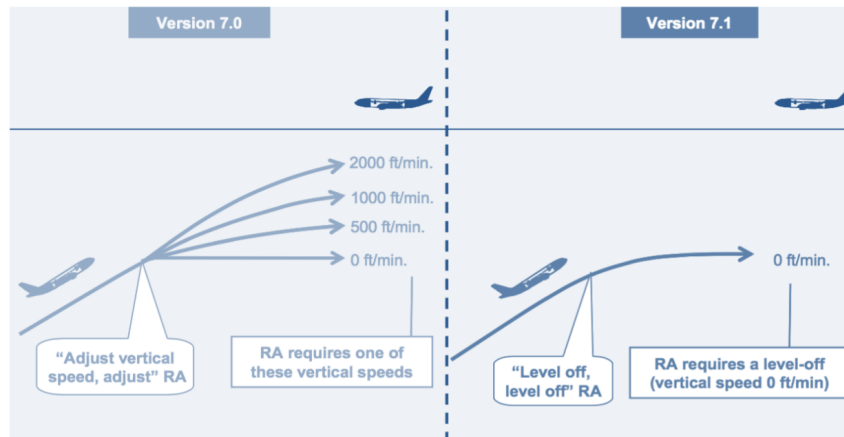


Figura 2.8 Comparación de las alertas sonoras "Adjust Vertical Speed, Adjust" y "Level off, Level off". Fuente: [2].

El objetivo del otro punto de mejora en TCAS II 7.1 era otorgar la capacidad de invertir o cambiar la primera instrucción dada por el aviso RA. Este detalle es de vital importancia para situaciones en las que una de las dos aeronaves involucradas no dispone de TCAS II, o cuando las instrucciones del ATC fuesen contradictorias con las del RA. De hecho, en julio de 2002 se produjo una colisión entre un Tupolev 154 y un Boeing 757 en Alemania por este motivo. Por ello, con la nueva versión, si se detecta que una aeronave no está haciendo lo que indica el RA, podrá emitirse uno nuevo a la que sí actúa en concordancia. El objetivo es que al cambiar su maniobra se pueda evitar la colisión. Para ello se podrán añadir a las alertas sonoras palabras como *NOW*, para que actúe de forma inminente. En la imagen 2.9 una de las dos aeronaves no actúa en concordancia al RA, y para solucionarlo se envía un nuevo aviso, concretamente "*Climb, Climb NOW*". De esta forma se evita la colisión que se iba a producir. [2]

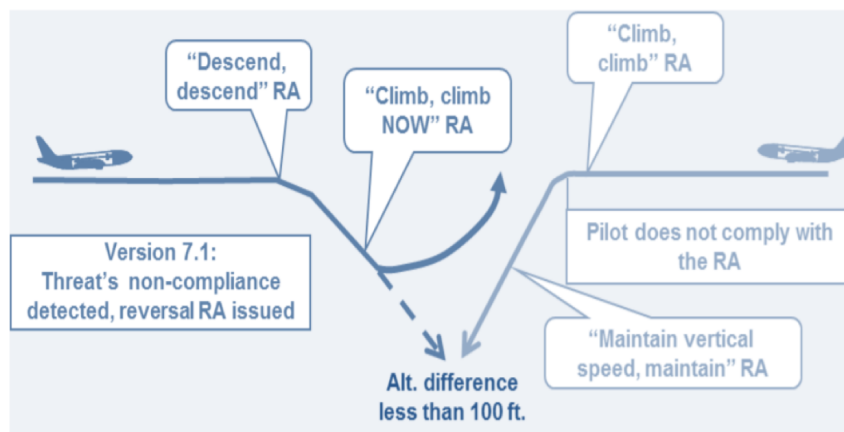


Figura 2.9 Mejora incluida en la versión 7.1, donde la aeronave equipada, evita la colisión gracias a un nuevo aviso RA. Fuente: [2].

En cuanto a la obligatoriedad del uso de estos equipos, según el Anexo 10 de ICAO [8] desde el 1 de enero de 2014, todas las nuevas instalaciones de ACAS II deben ser versión 7.1, y las ya existentes deberán actualizarse a la última versión antes del 1 de enero de 2017. Sin embargo, La Comisión Europea [3] adelantó estas fechas al 1 de diciembre de 2015 para todas aquellas aeronaves cuyo Maximum Certified Take-Off Mass (MCTOM) fuese mayor de 5700 kg o llevase más de 19 pasajeros, con la excepción de los vehículos aéreos no tripulados.

2.4.3 ACAS III

El sistema anticolidión ACAS III proporciona tanto avisos de tránsito como avisos de resolución en el plano vertical, y también en el horizontal. Por tanto, indica al piloto la velocidad vertical que se desea para evitar la colisión y el viraje que debe realizar. Sin embargo, la determinación de la posición horizontal es demasiado compleja para conseguir niveles de precisión adecuados. Ya que no se podía garantizar la seguridad, las dos implementaciones que se empezaron a desarrollar, TCAS III y TCAS IV, fueron abandonadas. Actualmente, no es obligatoria su instalación.

2.4.4 ACAS X

Desde 2008, la FAA ha financiado la investigación y el desarrollo de un nuevo sistema de aproximación empleando los recientes avances en la tecnología computacional. Uno de los avances incluidos es la programación dinámica, que consigue que la generación de alertas esté basada en una optimización de RAs off-line. ACAS X, cuyo nombre es temporal, parte de un modelo probabilístico de la posición futura de la aeronave. Mediante una tabla de búsqueda optimizada es capaz de detectar situaciones de peligro. La imagen 2.10 representa un esquema del funcionamiento interno de ACAS X. También emplea nuevas fuentes de navegación y vigilancia, basadas en satélites y tecnología ADS-B, al contrario que ACAS II que estaba basado en los transpondedores de a bordo.

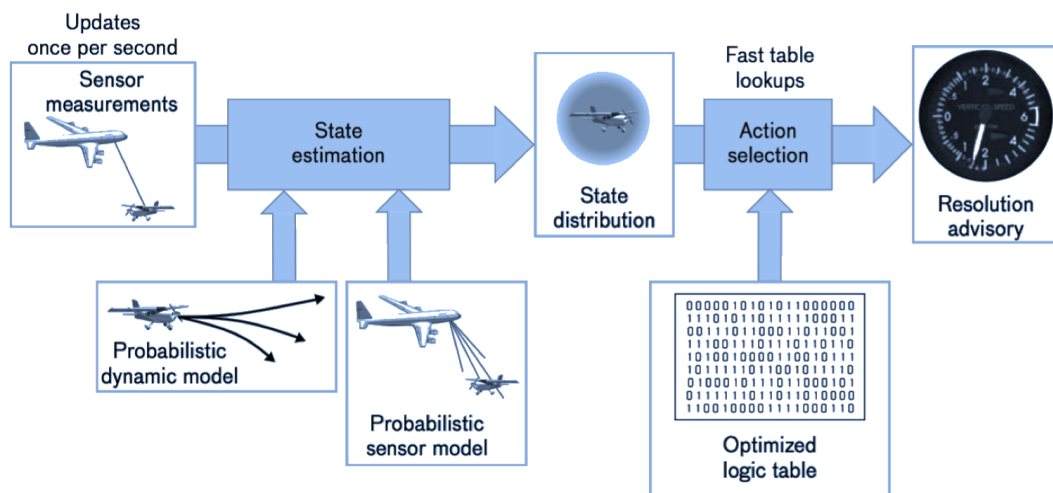


Figura 2.10 Funcionamiento interno de ACAS X. Fuente: [13].

Por todo ello, ACAS X incluye una serie de claros beneficios. Por un lado, una reducción de los avisos innecesarios, que tan molestos eran en el TCAS II. Tendrá capacidad para adaptarse a los conceptos operacionales futuros, ya que forma parte del plan Single European Sky Research (SESAR)⁵ y Next Generation Air Transportation System (NextGen)⁶. Además, lo más importante es que tendrá capacidad para extender la prevención de colisiones a aeronaves de tipo General Aviation (GA), con la variante ACAS X_A , y para UAV, con la variante ACAS X_U . Además, ACAS X proporcionará una mejora en la seguridad con unos cambios mínimos en hardware y RA, lo que facilitará la adaptación de los pilotos y los controladores. [9] [13] [22]

⁵ SESAR es un proyecto conjunto de la comunidad de transporte europea, que tiene como objetivo implantar una red Air Traffic Management (ATM) europea de altas prestaciones, para 2020.

⁶ NextGen es un proyecto de la FAA que busca una mejora en el entorno de los sistemas del espacio aéreo nacional, aumentando la seguridad y la eficiencia, para 2030.

2.4.5 Sistemas anticolidión en UAV

En el campo de la aeronáutica, al igual que en otros campos, los grandes avances tecnológicos surgen en el ámbito militar. Los sistemas aéreos no tripulados, Unmanned Aerial System (UAS), surgieron alrededor de los años 40, cuando en Inglaterra se desarrolló uno de las primeras aeronaves de combate no tripuladas, (Unmanned Combat Aerial System (UCAV)), a la que llamaron “*Killerbee*”. Posteriormente desarrollaron una unidad de observación, a la que llamaron “*Drone*”. Como anécdota, si se realiza una traducción literal, apodaron a su vehículo de combate “Abeja asesina”, mientras que al vehículo dedicado a observación lo llamaron “Zángano”.

Con el tiempo y los agigantados avances tecnológicos, los UAS no sólo son empleados en el área militar, si no que cada vez afectan más a la aviación civil. Recientemente han sido diseñados para realizar tareas muy concretas y complejas para los humanos. Son muchas las empresas que ya se dedican a la fabricación de UAVs, aunque algunos de los vehículos desarrollados sean simplemente elementos de ocio por control remoto. El problema que plantea el aumento de UAVs en el aire es la disminución de la seguridad para ellos mismos y para la aviación civil. Este hecho ha promovido el deseo de investigar, definir y desarrollar un sistema anticolidión para aeronaves no tripuladas, que permita volar con seguridad a todo tipo de aeronaves.

Desde el plan SESAR y NextGen, el ACAS X impulsa el prototipo de ACAS X_U dedicado a optimizar un sistema anticolidión para UAVs, como se mencionó anteriormente. Aún se encuentra en desarrollo y maduración, pero incluirá tecnología radar, infrarroja y electro-óptica, entre otras, y proporcionará avisos verticales y horizontales. [22]

Sin embargo, no sólo se está impulsando a nivel europeo, si no que su investigación y desarrollo está en alza estos últimos años. Algunos investigadores consideran que el desarrollo de un sistema anticolidión para vehículos no tripulados debería basarse en un algoritmo de fácil implementación llamado Velocity Obstacle (VO). [12] Incluyendo ciertas modificaciones que lo hacen compatible con UAVs, surge un método llamado Selective Velocity Obstacle (SVO). Para esta propuesta es importante hacer distinción entre aeronaves cooperativas y no cooperativas⁷, puesto que las Reglas del Aire recogidas en el Anexo 2 de ICAO [10] serán diferentes.

Por un lado, para aeronaves cooperativas, las reglas del aire siguen el siguiente criterio:

- En encuentros convergentes, el vehículo que se encuentre a la derecha tiene preferencia.
- En aproximación frontal, ambos UAV deberán desplazarse hacia su derecha.
- En adelantamiento, el que va a ser adelantado debe permanecer a su derecha.

Con estas reglas surge un sistema anticolidión para aeronaves cooperativas, llamado Co-ACAS.

En el caso de aeronaves no cooperativas, básicamente no hay reglas. Se trata de evitar de la forma más rápida posible, en las tres dimensiones, en cuanto el objeto o el conflicto sea detectado. Surge así una nueva visión en la anticolidión, conocida como “*Sense and Avoid*”.

Otros autores [15], proponen emplear el concepto “*Learning from Demonstration*”. El objetivo es definir un comportamiento basado en las maniobras que realizaría el ser humano, para así entrenar a la aeronave no tripulada de forma que presente las mismas capacidades de prevención que posee una aeronave de la aviación civil.

En [1] lo que se propone es crear un mapa tridimensional de posibles colisiones futuras partiendo de los datos de la aeronave y de la información de las inmediaciones. Una vez detectada una colisión, se deberá

⁷ Aeronave cooperativa: aquella equipada con los sistemas necesarios para detectar y ser detectada.

Aeronave no-cooperativa: aquella que no dispone de los sistemas necesarios para indicar a las aeronaves vecinas su posición, altitud, etc.

calcular el tiempo necesario para maniobrar; cuando éste sea cero, se realizaría una maniobra automática de escape que conduciría a la aeronave a una situación fuera de peligro.

A pesar de todos los prototipos que se han ido investigando, la mayoría sigue en fase de desarrollo o están siendo sometidos a ensayos, hasta conseguir que los niveles de seguridad sean los aceptados. Otros muchos también han sido desestimados por no poder lograr su cometido.

Queda en evidencia la necesidad de nuevos sistemas que aseguren la no colisión en el espacio aéreo, ya se trate de aeronaves tripuladas o no tripuladas. Es importante aportar un nuevo enfoque a estos sistemas, proporcionando una alternativa o información complementaria. Por ejemplo, una buena opción sería aprovechar la cantidad de información que hoy día se puede extraer de una imagen. La computación digital de imágenes ya ha permitido enormes avances en otros campos de la ciencia, como la medicina. Esto, unido a la visión estereoscópica puede proporcionar información sobre la localización de aeronaves vecinas, sin necesidad de que éstas estén equipadas con ningún sistema específico. Así, se dispondría de un sistema anticolisión compatible con aeronaves de aviación civil, militar, general y no tripuladas.

3 Procesamiento digital de imágenes

El ser humano, en su vida diaria, emplea la visión para conocer la posición, el tamaño y demás características de cada uno de los objetos que le rodean. Esto le permite identificar cada uno de ellos y desempeñar todo tipo de actividades. Por ejemplo, es capaz de desempeñar tareas inteligentes, como la investigación científica o el control de sistemas, y otras muchas tareas de la vida cotidiana, como el simple hecho de caminar.

Con el continuo avance de la tecnología y el avance en otras áreas, como la computación y las matemáticas, ha aumentado el desarrollo de los sistemas de visión artificial. Con ellos se pretende que a través de una o varias imágenes se pueda detectar y reconocer el contenido de ellas. Todo ello, ligado a una simulación de la visión humana por parte de los ordenadores, da lugar a lo que se conoce como “visión computacional”. Estos avances son de utilidad para numerosos campos de la ciencia, como la medicina, la astronomía o la geología, y son de aplicación en la vida diaria, como por ejemplo la información meteorológica.

Los métodos de procesamiento de imágenes tienen dos objetivos. Por un lado, la mejora de la información que contienen las imágenes para la correcta interpretación humana. Por otro lado, el posterior tratamiento de los datos que proporciona la imagen a nivel de computación.

En los siguientes apartados, se tratará de explicar qué es una imagen a nivel de datos, y como es posible procesarla para la detección de objetos.

3.1 Espacios de color

En el procesamiento de imágenes en color se necesita emplear un determinado modelo de color. Este espacio está definido por un sistema de coordenadas tridimensional y un subespacio que pertenece a este sistema. De esta forma, cada color está representado por un único punto, quedando definido cuantitativamente de forma unívoca.

De entre todos los posibles espacios de color que existen, se ha elegido trabajar en el modelo Red Green Blue (RGB), ya que no requiere de ninguna transformación para ser empleado en el procesamiento digital de imágenes. Además, es el formato estándar que incluyen la mayoría de las cámaras y las pantallas a color. Este modelo se basa en un sistema tridimensional en coordenadas cartesianas, donde cada componente corresponde a uno de los colores de luz primarios, rojo, verde y azul. El esquema del modelo está representado en la imagen 3.1.

En una imagen RGB cada color se representa por la adición de varias intensidades. Concretamente, está compuesto por las intensidades que contiene de cada uno de los colores primarios. En la herramienta que se va a emplear, *Matlab*, esto se traduce en que una imagen RGB está formada por 3 matrices. Las dimensiones de dichas matrices se corresponderán a la resolución de la imagen. La combinación del mismo elemento en

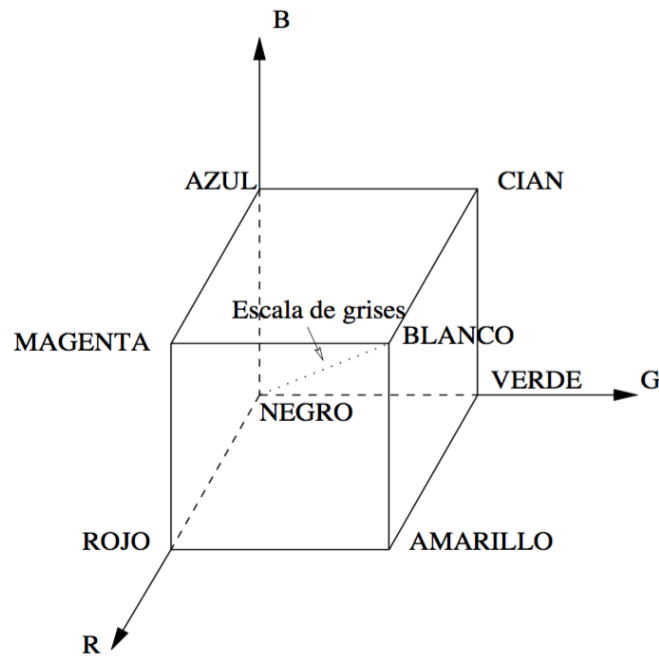


Figura 3.1 Espacio de color RGB. Fuente: [5].

cada una de las matrices corresponde a un sólo pixel, que es la unidad mínima de información. [5] La imagen 3.2 muestra el esquema matricial planteado anteriormente.

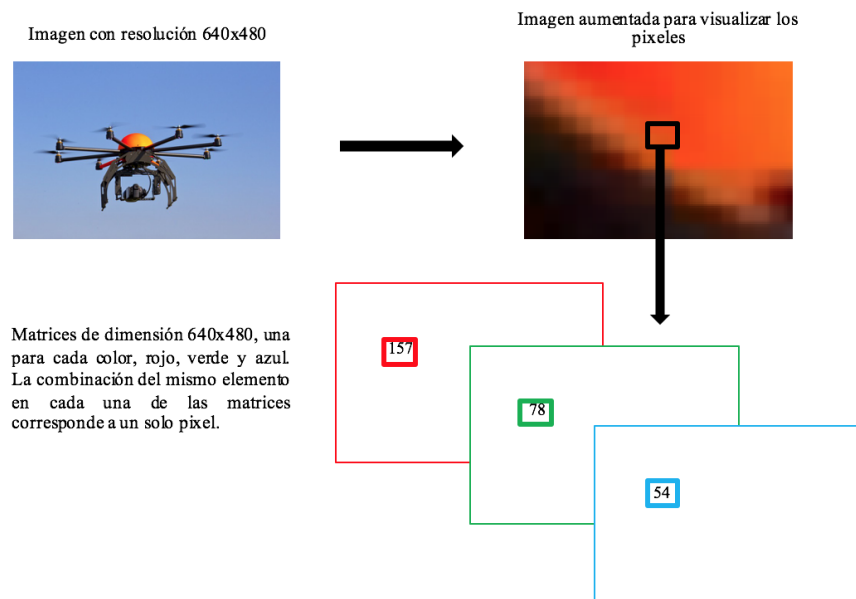


Figura 3.2 Imagen RGB compuesta por tres matrices de la misma dimensión.

De esta forma, el tratamiento de las imágenes en *Matlab* se reduce a trabajar con tres matrices concatenadas, o con tres matrices independientes, cada una correspondiente a un plano de color primario.

3.2 Detección de objetos

En el proceso de detección de objetos en la imagen, se trata de realizar un análisis de las características de cada objeto, con el fin de identificarlo y clasificarlo. A través de la imagen, se obtiene una proyección bidimensional (2D), de un objeto tridimensional (3D). De esta proyección es posible extraer todo tipo de características, como su color, forma o textura, lo que se conocen como descriptores.

3.2.1 Descriptores de imagen

La información contenida en una imagen se digitaliza como el valor de cada uno de los píxeles. Con ellos, se establece la unión entre el contenido de una imagen real y su valor abstracto. Ya que la información que proporciona una imagen debe ser analizada de forma objetiva y automatizada, surge la idea de emplear descriptores de imagen.

Un descriptor ideal debería definirse por las siguientes propiedades [6]:

- **Simplicidad.** Debe ser capaz de proporcionar de manera sencilla y clara las características de la imagen, para que su contenido sea fácilmente interpretable.
- **Invarianza.** Debe proporcionar robustez para relacionar elementos que presenten cualquier tipo de deformación de una imagen a otra.
- **Eficiencia.** Se desea que un descriptor proporcione los resultados esperados con prontitud y sin necesidad de mucho espacio, con el fin de ser empleados en aplicaciones que tengan poco margen de espacio y/o tiempo.
- **Repetibilidad.** Aquel descriptor que se genere a partir de una imagen, será el mismo independientemente del instante en el que se genere.
- **Diferenciabilidad.** El descriptor generado tiene que poseer gran capacidad de discriminación respecto de otras imágenes, a la vez que debe ser capaz de relacionar imágenes similares.

3.2.2 Clasificación de los descriptores

Según el nivel de abstracción en la representación, los descriptores se pueden clasificar en dos grupos diferenciados :

- **Descriptores de información general:** en un nivel bajo de abstracción se engloban los descriptores visuales, que describen características básicas presentes en la imagen, como el color, las formas, las regiones, las texturas y los movimientos.
- **Descriptores de información específica:** en un nivel más alto, los también llamados descriptores semánticos, aportan la información referente a los objetos y situaciones que aparecen en la escena, apoyándose para ello en los descriptores más generales.

Dentro de los descriptores de bajo nivel, se puede hacer una clasificación en función de las regiones de la imagen sobre las que actúen. De esa forma se tienen dos niveles más de clasificación [6]:

- **Descriptores Globales.** Comprimen el contenido de una imagen en una única matriz (o vector) de información. Son capaces de reunir gran cantidad de información en una pequeña porción de datos con los que detallarla. Por ello, debido a su bajo costo computacional y su simplicidad, este tipo de descriptores son muy utilizados.

- **Descriptores Locales.** Estos descriptores actúan en regiones anteriormente identificadas, que contienen información de interés. Construyen un vector o matriz de información para cada región de interés, y para las regiones vecinas, de forma que el descriptor completo está formado por todos los vectores de características. Estas regiones se conocen como puntos de interés, y suelen referirse a pequeñas porciones de la imagen, o bordes concretos.

En la imagen 3.3 se muestra un ejemplo de comparación entre descriptores globales, como el histograma de color, y descriptores locales, como el descriptor SIFT.

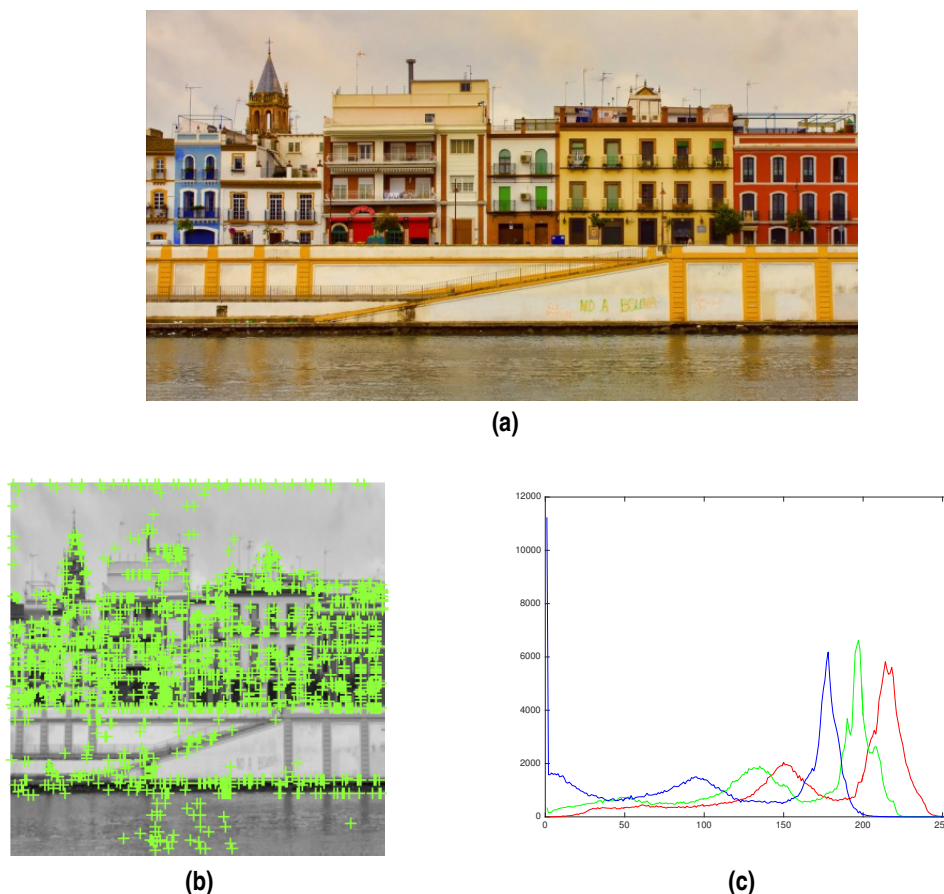


Figura 3.3 Comparación entre detector local, b), y detector global, c).

3.2.3 Descriptores de color

Tanto para la visión humana como para el procesamiento de imágenes, el color es una característica básica pero importante, con la que se muestra información relevante sobre los objetos. Existen concretamente cinco herramientas que sirven como descriptores basados en el color. Tres de ellas se centran en la distribución del color, que son Dominant Color Descriptor (DCD), Color Structure Descriptor (CSD) y Scalable Color Descriptor (SCD); las dos restantes se basan en la distribución espacial del color, Color Layout Descriptor (CLD), y en la relación del color entre un conjunto de imágenes, Group of Frame (GoF). Se explican a continuación las 3 consideradas más relevantes, el DCD, el CSD y el CLD. [25] [14] [24]

Dominant Color Descriptor

El DCD es un descriptor de color muy compacto. Extrae las características de la imagen a partir de la agrupación de todos los colores que contiene la imagen en una pequeña cantidad de ellos. Con ello es capaz

de obtener cuáles son los colores dominantes, sus porcentajes y muchos otros parámetros. Se define como:

$$F = \{(c_i, p_i, v_i), s\}, \quad (i = 1, \dots, N) \quad (3.1)$$

donde c_i representa los colores, p_i sus porcentajes, v_i la opción de variantes para cada color dominante, s la posible coherencia espacial de su color dominante, es decir, la homogeneidad global espacial, y N el número máximo de colores dominantes. [25]

Cada color dominante, c_i , es un vector cuyas componentes se corresponden con el espacio de color (en el presentado anteriormente sería un vector 3-D en el espacio de color RGB). Se pueden encontrar un máximo de 8 colores dominantes para representar una imagen o región de la imagen. A este número hay que incluir una cierta permisibilidad en el valor del pixel de color, en caso de tener en cuenta la variabilidad v_i . El porcentaje p_i , cuyo valor está normalizado entre 0 y 1, representa el número de píxeles que hay en la imagen de color c_i , y por tanto, $\sum_i p_i = 1$. Con el parámetro s se pretende identificar los grupos de píxeles con el mismo color dominante que se encuentran conectados espacialmente.

En la imagen 3.4 se puede observar el resultado de aplicar el descriptor DCD con el máximo de colores dominantes, en este caso 8.



(a) Imagen original.



(b) Imagen modificada .

Figura 3.4 Aplicación del descriptor DCD con 8 colores dominantes.

Durante la extracción de características, se aplica a la imagen el Algoritmo de Lloyd Generalizado, [19], como algoritmo de agrupamiento, cuyo objetivo es minimizar la distorsión de cada punto escogido i . Para

ello, el algoritmo parte de 8 centroides asignados de forma uniforme en el espacio de color, que no es más que 8 agrupaciones distintas dentro del espacio de color. Entonces, a cada punto se le va asignando el centroide más cercano cumpliendo cierta distorsión mínima.

Se trata de un descriptor simple y robusto, lo que constituye una ventaja frente a otros descriptores de color. Además presenta un bajo costo computacional. Como inconveniente, el resultado de esta implementación podría ser incompleto debido a la forma de elegir los centros iniciales. La solución sería asignar a los puntos iniciales cada uno de los 8 posibles colores dominantes.

Color Structure Descriptor

Este descriptor esta basado en un histograma de color, al que añade cierta información espacial. Es de utilidad para aquellas imágenes que tienen el mismo histograma de color pero son distintas entre sí, como ocurre en la imagen 3.5. Su objetivo es proporcionar una descripción que mejore la identificación de las distribuciones de color. Para ello, CSD se caracteriza por un histograma estructurado de color, con M colores cuantificados, c_m , y se expresa como sigue:

$$h(m), \quad m = 1, \dots, M \quad (3.2)$$

donde $M \in \{256, 128, 64, 32\}$ y el valor de $h(m)$ es el número de elementos estructurales que contienen uno o más píxeles de color c_m . Un elemento estructural no es más que cada una de las partes diferenciadas, aunque vinculadas, en las que se puede dividir un espacio. El histograma estructurado se extrae de la imagen por acumulación de pequeñas ventanas de 8x8, los elementos estructurales. A continuación se escanea la imagen y se cuenta el número de veces que un color en concreto aparece en un elemento estructural. De esta forma, el valor del histograma irá aumentando progresivamente. Se denota ahora I como el conjunto de todos los colores cuantificados en la imagen, y $S \in I$, como el conjunto de colores en una subsección de la imagen. Mientras el elemento estructural permanezca escaneando la imagen, el valor del histograma irá variando siguiendo la siguiente expresión:

$$h(m) \leftarrow h(m) + 1, \quad m \in S \quad (3.3)$$

Así, el valor final de $h(m)$ se determina con el número de posiciones en las cuales el elemento estructural contiene c_m . [25]

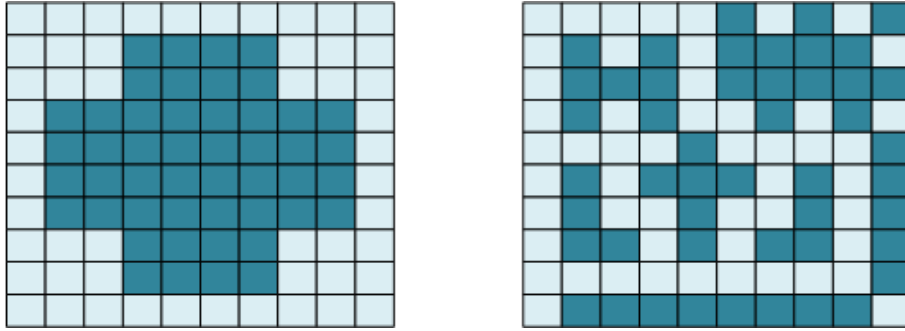


Figura 3.5 Imágenes diferentes cuyo histograma es el mismo.

Como ventaja, CSD proporciona una representación con alta precisión, que es similar a la percepción visual del humano; como desventaja, requiere gran espacio para ser usado. El mínimo número de bytes con los que puede trabajar es 32, que comparando con el descriptor DCD, habrá situaciones en las que incluso lo triplique.

Color Layout Descriptor

Este descriptor obtiene la distribución espacial de los colores representativos en la imagen o en una región de ella. La representación se basa en la Transformada de Coseno Discreta (Transformada de Coseno Discreta). Se trata de una transformada basada en la Transformada de Fourier Discreta, pero empleando únicamente

números reales.

El proceso de extracción del descriptor en la imagen contiene 4 partes: la división de la imagen, la selección del color más representativo, la transformada Transformada de Coseno Discreta y la exploración de los coeficientes obtenidos mediante escaneo en zig-zag. En la primera etapa, la imagen de entrada se divide en 64 bloques para garantizar la invariabilidad de la resolución; pero no todas las imágenes tendrán una resolución tal que permitan obtener 64 bloques de igual dimensión. Por ello, este número admite cierta flexibilidad, siempre y cuando se siga una distribución de píxeles lo más uniforme posible. En la etapa inmediatamente después, se selecciona un único color como representante de cada bloque. No se emplea ningún complejo método, simplemente haciendo uso de la media del color de los píxeles contenidos es suficiente. En la tercera etapa, se obtienen tres grupos de 64 coeficientes Transformada de Coseno Discreta, cada uno correspondiente a las 3 componentes de color de cada pequeño bloque. En la última etapa, cada conjunto de coeficientes se escanea siguiendo una trayectoria zig-zag y una pequeña cantidad de coeficientes de baja frecuencia serán cuantizados. Finalmente se obtiene un conjunto de tres matrices correspondientes a los CLD de la imagen de entrada. El esquema seguido en el proceso se muestra en la imagen 3.6.

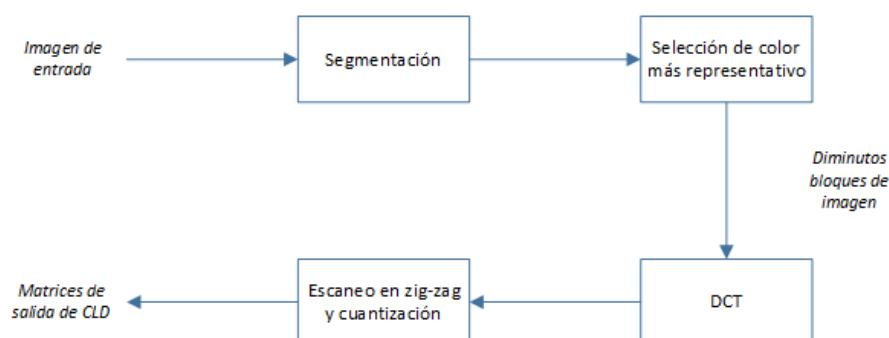


Figura 3.6 Proceso seguido por el descriptor CLD.

Con este descriptor se consigue evaluar si dos imágenes son similares o no. Se trata de un descriptor muy eficiente y muy compacto, además de rápido en aplicaciones de búsqueda.

3.2.4 Descriptores de textura

En una imagen, la textura constituye una importante característica visual para búsqueda y diferenciación entre una gran cantidad de imágenes con un patrón similar. Aunque el concepto de textura sea ampliamente conocido y entendido, no existe ninguna definición que sea aceptada a nivel universal. Un descriptor de textura se basa en la homogeneidad de las regiones presentes en una imagen y los histogramas de bordes de las regiones. Entre el conjunto de descriptores se encuentran Homogeneous Texture Descriptor (HTD), Texture Browsing Descriptor (TBD) y Edge Histogram Descriptor (EHD). [14]

- **Descriptor de Textura Homogénea.** Este descriptor proporciona una caracterización cuantitativa de las regiones con textura homogénea. A partir de ahí, extrae las similitudes analizando imagen por imagen. Para ello, se basa primero en un filtrado de la imagen con diversos filtros y, posteriormente, analiza la desviación obtenida a la salida de los filtros en el dominio de la frecuencia. Durante este filtrado suele usarse el filtro de Gabor, que se trata de un filtro lineal cuya respuesta está multiplicada por una función gaussiana, lo que hace que las Funciones de Gabor se localicen tanto en el dominio frecuencial como en el dominio espacial. Además, se ha relacionado el filtrado de una imagen con funciones de Gabor con los procesos en la corteza visual, y sirven como modelo en la investigación de las células simples de la corteza cerebral relacionadas con la recepción visual. Por ello, aunque se trate de un descriptor de fácil implementación y muy robusto, es muy eficiente en el procesamiento digital de imágenes. [20]
- **Texture Browsing Descriptor.** Se trata de un descriptor muy compacto, que requiere únicamente 12

bits como máximo para caracterizar la regularidad en textura (2 bits), su direccionalidad (3bits x 2) y su dimensión (2 bits x 2). El filtrado de la imagen y el posterior análisis se realiza de forma similar al descriptor HTD.

- **Descriptor de Histograma de Borde.** Obtiene la distribución espacial de los bordes, que constituye una buena muestra de la textura cuando se analiza imagen a imagen, incluso cuando la textura fundamental no es homogénea. Para su análisis, la imagen inicial se va subdividiendo en imágenes más pequeñas hasta lograr bloques de 2x2. Entonces, se aplica el descriptor a ese bloque como si fuese un solo pixel, con la media de la intensidad como valor de intensidad del bloque.

3.2.5 Descriptores de forma

Los descriptores de forma poseen una información muy importante para identificar un objeto y su funcionalidad, e incluso puede ser usados para el reconocimiento de objetos. Los humanos somos capaces de reconocer objetos simplemente por su forma; sin embargo, para que esta información sea de utilidad en computación, deben fraccionarse las imágenes de manera similar a como lo hace el sistema visual humano. Puesto que este sistema es muy complejo, actualmente solamente se dispone de una serie de algoritmos que permiten realizar una buena aproximación. Estos descriptores definen las regiones, formas y contornos de objetos 2-D y los volúmenes de objetos 3-D.

Dos descriptores capaces de obtener diferentes características morfológicas en imágenes 2-D son:

- **Region-based Shape Descriptor (RSD):** obtienen la distribución de todos los píxeles dentro de una región.
- **Contour-based Shape Descriptor (CSD):** describen las propiedades de la forma del contorno de un objeto.

3.3 Detección de borde

En ocasiones, en lugar de la detección de objetos basada en descriptores, se realiza una detección de bordes. De esta forma, a partir de los contornos o fronteras localizados, se es capaz de extraer los objetos contenidos en una imagen.

Cualquier detector de bordes debe recibir como entrada una imagen digital, y la transformará en un mapa de bordes. Algunos detectores añaden en estos mapas información sobre la orientación, posición o fuerza de los bordes detectados.

Cuando se hace referencia al concepto borde, generalmente indica una discontinuidad en alguna propiedad de la imagen. En la mayoría de las situaciones, por simplicidad, se emplean imágenes monocromáticas, donde existe una variación en los niveles de gris. En dichas imágenes, los bordes se corresponderán con saltos o cambios en la intensidad luminosa de los niveles de gris. La forma del borde dependerá por tanto de la función de intensidad, permitiendo distinguir varios tipos de borde según su forma. Concretamente son el borde en escalón, rampa, escalera, pico, valle, tejado, pico o pulso. Los perfiles de la función de intensidad a lo largo de una línea perpendicular al borde que se desea detectar se muestran en la figura 3.7. [5]

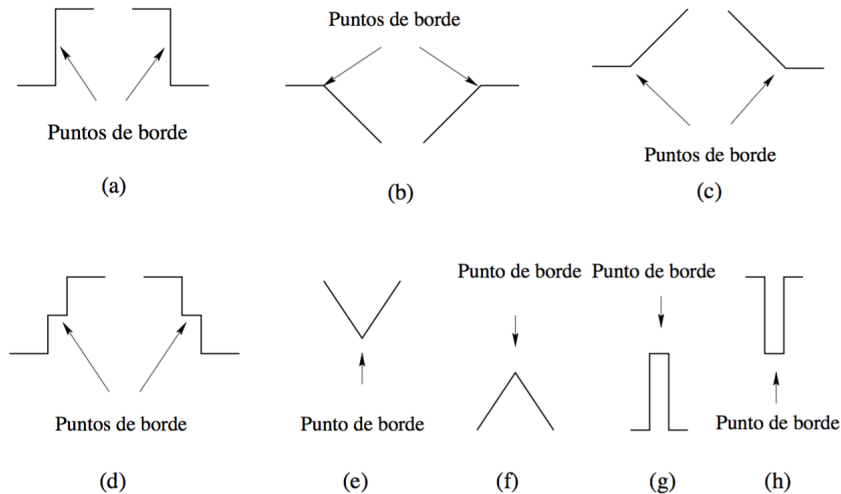


Figura 3.7 Formas posibles del borde en una imagen. *Fuente:* [5].

El borde más comúnmente detectado es el borde en escalón, que aparece al coincidir dos zonas homogéneas con niveles de gris muy distintos entre sí. El borde se encuentra, por tanto, en el lugar en el que se produce el cambio brusco de intensidad. Suele aparecer cuando hay una sombra, o el objeto se encuentra parcialmente oculto. Desde un punto de vista geométrico, los bordes pueden ser líneas rectas, con cualquier orientación, líneas curvas, esquinas o uniones. Estas últimas pueden tener forma de T, L, Y o X, y se producen cuando dos bordes o más coinciden en algún punto. En la imagen 3.8 se puede ver el ejemplo de un borde del tipo T.

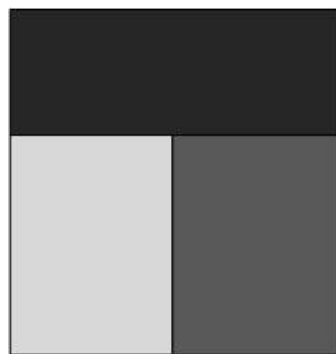


Figura 3.8 Borde del tipo T.

3.3.1 Algoritmos de detección de bordes en imágenes monocromáticas

En paralelo al desarrollo del procesamiento digital de imágenes, se han ido desarrollando los algoritmos de detección de borde. Teniendo en cuenta que los detectores de borde deben formar parte de un sistema de visión, se pueden clasificar en detectores autónomos o contextuales. Los autónomos no requieren de información previa ni de la ayuda de ningún otro componente del sistema de visión en el que se encuentran integrados. Realizan un procesamiento local, basado en el entorno. Los detectores contextuales conocen previamente la imagen que va a ser procesada, por lo que requieren datos proporcionados por otros componentes. Para sistemas de visión más generales son apropiados los primeros, mientras que para imágenes específicas con objetos específicos resultan más adecuados los segundos.

La algoritmia empleada en la detección de bordes en imágenes monocromáticas está basada en derivadas de primer orden (como por ejemplo métodos basados en gradientes), en derivadas de segundo orden (como

el laplaciano) y en criterios de optimización.

Considerando una imagen como una función continua $f(x,y)$, en las zonas en las que exista un cambio brusco de intensidad en el nivel de gris, su derivada tendrá un máximo local en esa dirección, que será la del contorno. Por ello, el operador gradiente es muy empleado cuando las imágenes que se procesan presentan claras transiciones de intensidad y ruido relativamente bajo.

En cuanto a las fases llevadas a cabo en la detección, se incluyen tres operaciones fundamentales: [5]

- **Suavización de la imagen.** Su función es la de reducir el ruido para obtener una buena detección. En esta primera etapa se deberá tomar una solución de compromiso, puesto que realizar una suavización de la imagen a gran escala podría ocasionar la pérdida de información, mientras que si se hace a pequeña escala, podría no eliminar por completo el ruido.
- **Diferenciación de la imagen.** Resalta las propiedades de los bordes a partir de la evaluación de las derivadas de la imagen. Con esta fase lo que se consigue es hacer más evidente la discontinuidad que se produce cuando hay un borde, y de esta forma, se facilita la extracción de las características de los bordes.
- **Identificación de los bordes.** Identifica los bordes falsos y los elimina, para quedarse así con los que son reales. Con la supresión de los bordes falsos aumenta la proporción Signal-to-Noise Ratio (SNR) en la imagen obtenida tras las dos fases anteriores.

A partir de la imagen de entrada, y tras la realización de las tres fases anteriores, se obtiene el mapa de bordes deseado. Estas tres fases están estrechamente relacionadas, de forma que el correcto desarrollo de una de las operaciones repercute en las posteriores. En general, no hay forma intuitiva de conocer si los píxeles que componen los bordes detectados son correctos o no. Es lo que se conoce como falsos positivos o falsos negativos, en función de si devuelve un pixel cuando no debe o no lo devuelve cuando sí debería haberlo hecho. Para la detección correcta son necesarios otros mecanismos, como los de enlazado de bordes, de crecimiento de regiones o de umbralización.

El enlazado de bordes consiste en analizar los atributos de los vecinos de cada pixel, de manera que se realiza un procesamiento local para detectar puntos que sean similares entre sí. Todos estos puntos similares se enlazan. Para considerar la similitud entre los píxeles del borde se analiza la intensidad del gradiente y su dirección.

En el crecimiento de regiones se agrupan los píxeles en regiones cada vez mayores siguiendo un cierto criterio. Normalmente, se parte de unos puntos denominados “semillas”, y se van añadiendo aquellos píxeles vecinos que cumplan ciertas condiciones. Hay que tener en cuenta las condiciones de conectividad o adyacencia de la imagen, pudiendo ser conectividad a 4⁸ o a 8⁹, de forma que el número de regiones variará.

En cuanto a la umbralización, se trata de un método básico que permite detectar un objeto del fondo de la imagen a través de una binarización de la misma. Para ello, se obtiene el histograma de la imagen, que proporciona el número de píxeles que tienen un cierto nivel de gris. Eligiendo un umbral adecuado de forma que el histograma adopte una forma de valle en torno a ese valor, todos los niveles de grises que queden por debajo del umbral serán negro y todos los mayores blanco.

Por último, cabe destacar la necesidad de tener un contorno cerrado en la detección de borde. Para ello se realiza un algoritmo de seguimiento de contorno imponiendo cierta condición que puede ser referente al gradiente, por ejemplo. Un ejemplo de algoritmo sería el siguiente:

1. Se comienza dentro de la región.

⁸ Conectividad a 4: considera la conectividad horizontal y vertical.

⁹ Conectividad a 8: considera la conectividad horizontal, vertical y diagonal.

2. Si el pixel pertenece a la región, se gira a la izquierda y se salta de pixel; si no, se gira a la derecha y se salta de pixel.
3. Se continua hasta que se llegue al pixel de inicio.

El camino que seguiría el algoritmo se muestra en la imagen 3.9. De esta forma se obtiene un contorno cerrado aunque la región no lo sea.

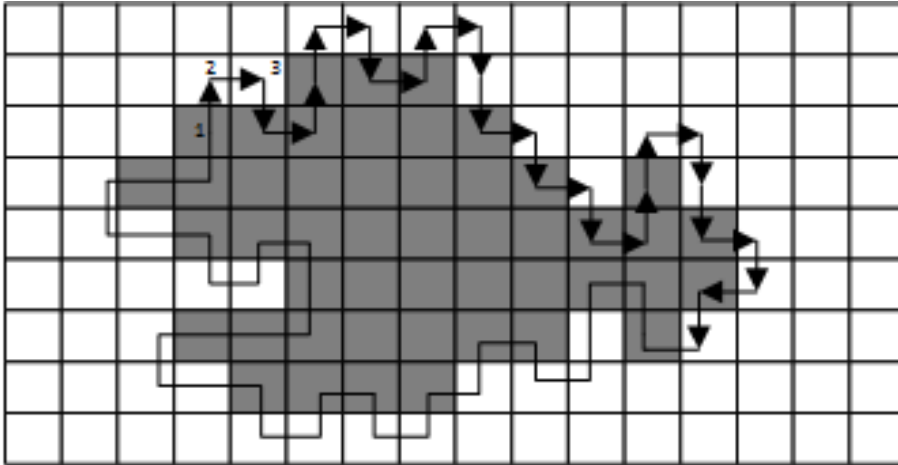


Figura 3.9 Ejemplo de seguimiento de contorno.

4 Desarrollo del proyecto y resultados

En este capítulo se presenta la principal aportación de este Trabajo Fin de Grado. En él se presentan todos los pasos desarrollados hasta lograr el objetivo último de calcular la distancia a un objeto.

Los primeros pasos fueron la familiarización con el procesamiento digital de imágenes. Como se vio en el capítulo anterior, dentro de una imagen se puede hacer tanto detección de borde como detección de objetos al completo. Para ambas detecciones se ha empleado el color como descriptor.

Puesto que el cálculo de distancia estará basado en visión estereoscópica a partir de dos imágenes, será necesario realizar una identificación previa de los objetos en ambas imágenes. Posteriormente, estaremos en disposición de calcular la distancia.

4.1 Detección de borde

La detección de borde se trata de un paso básico y fundamental en el procesamiento digital de imágenes y la extracción de información de las mismas. Para la obtención de los bordes se analizan las variaciones bruscas que se produce en alguna propiedad.

Una imagen en escala de grises puede tratarse como una función bivariable $f(x,y)$, donde x e y indican la posición del pixel. En este caso, se trata por tanto de analizar la variación que se produce en la intensidad, que corresponderá con las fronteras de los objetos. Por tanto, se partirá de una imagen a color que se transformará a escala de grises mediante *Matlab*. Algunos de los comandos más importantes empleados son:

- » *imread()* : Lectura de la imagen
- » *rgb2gray()* : Transforma la imagen a escala de grises
- » *imshow()* : Muestra la imagen por pantalla

Una forma sencilla de obtener los bordes en una imagen en escala de grises es la aplicación del gradiente. Se recuerda que el gradiente indica la dirección en la que la variación es máxima, y que se expresa como indica (4.1).

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (4.1)$$

El objetivo será entonces localizar los píxeles donde la magnitud del gradiente (4.2) se hace máxima.

$$|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \quad (4.2)$$

La herramienta utilizada en la implementación del proyecto, *Matlab*, tiene incluida una función que calcula el gradiente de la imagen. Dicha función es »*gradient()*. Aplicándola a la imagen a analizar se obtiene una primera aproximación a los bordes de la imagen. El resultado se muestra en la imagen 4.1.

Puesto que la detección de bordes es un campo muy estudiado, existen ya ciertos algoritmos de detección reconocidos. La mayoría se basan en la aplicación del gradiente a la imagen. Los más importantes incluso están implementados en las funciones internas de *Matlab*.

Uno de ellos es el algoritmo de Canny, que está considerado como uno de los mejores. Fue desarrollado por un catedrático de la universidad de Berkeley en 1986, John F. Canny. El objetivo de Canny era desarrollar un detector de bordes óptimo. Para ello, el detector debía cumplir tres criterios: un criterio de detección, para que no cree falsos positivos ni falsos negativos; un criterio de localización, para que los bordes estén bien situados; y un criterio de una respuesta, de forma que los bordes sean detectados de forma unívoca.

El algoritmo consta de tres etapas. El primer paso es la obtención del gradiente, calculando tanto su magnitud como su orientación. Es necesaria la aplicación de un filtro gaussiano con una desviación estándar que suavice la imagen y reduzca el ruido, pero siempre con una solución de compromiso que no elimine información. En la siguiente etapa se aproximan las direcciones obtenidas de los gradientes, de forma que sólo se tengan cuatro valores respecto la horizontal, 0° , 45° , 90° y 135° . Posteriormente se observa el valor de la magnitud de sus vecinos. Si es menor se le asigna un cero; si no, el valor que tuviera la magnitud. De esta forma se obtiene una imagen con los bordes adelgazados. Puesto que esta imagen suele contener falsos bordes detectados por el ruido, el último paso será la eliminación de estos bordes mediante umbralización. Es importante ahora realizar un cerrado de los contornos que hayan podido quedar abiertos por la presencia de ruido. Para encontrar los posibles contornos no cerrados, se realiza un seguimiento de la dirección máxima del gradiente. [17]

Para su implementación en *Matlab* basta con emplear la función »*edge()* y especificar en sus argumentos de entrada el tipo de algoritmo a emplear, en este caso Canny. Los resultados se muestran en la imagen 4.1.

Otro algoritmo muy empleado es el algoritmo de Sobel. En este caso, el algoritmo calcula el gradiente para cada pixel, y obtiene su magnitud y dirección. Los argumentos de salida que ofrece los obtiene comparando cada pixel con sus vecinos, horizontal o verticalmente. De esta forma, devuelve la magnitud del mayor cambio posible, la dirección en la que tiene lugar y el sentido de oscuro a claro. Así, si se trata de una zona en la que no hay cambio de intensidad, el algoritmo devolverá un vector nulo. Si por el contrario se trata de un contorno, devuelve un vector que lo cruza perpendicularmente, de más oscuro a más claro.

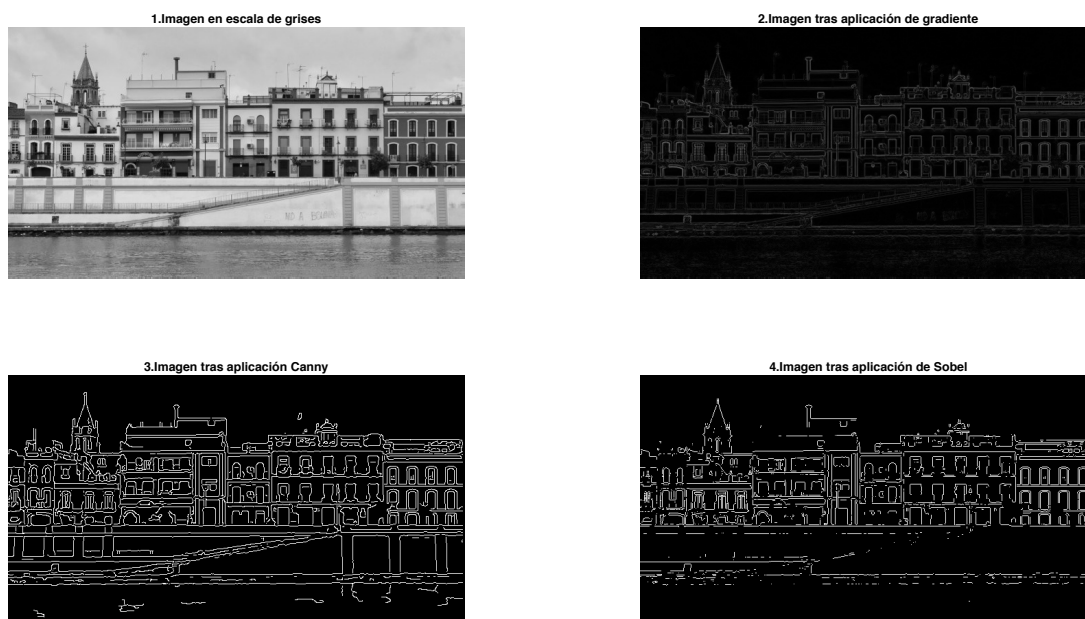


Figura 4.1 Imagen comparativa de detección de borde.

Para su implementación, se emplea la misma función anterior `»edge()` y se especifica el tipo de algoritmo a emplear, en este caso Sobel.

En la imagen 4.1 se observa una comparación entre la detección de borde realizada con los tres métodos planteados anteriormente. Es evidente que los algoritmos de Canny o Sobel son mucho mejores que la aplicación mera del gradiente. Pero si se observa con más detenimiento, como se había anticipado, el operador de Canny ofrece mejores resultados, proporcionando contornos cerrados y un número mayor de contornos detectados.

Como aportación a la detección de borde, y con el fin de conocer en profundidad cómo funcionan estos algoritmos, se plantea el desarrollo de un algoritmo propio de detección. El objetivo es que sea capaz de detectar el contorno de objetos en imágenes. Para ello, como se ha visto anteriormente, será necesario encontrar alguna discontinuidad en alguna propiedad de la imagen. Dicha propiedad puede ser cualquiera, como la forma, la textura y el color. Analizar la forma supone una dificultad añadida, puesto que los objetos se desfiguran en función de la posición a la que se extraigan las imágenes de entrada. Sin embargo, el color es una propiedad que se mantiene bajo unas mismas condiciones de iluminación y entorno. Por la simplicidad que ofrecía, se escogió el color como propiedad a analizar.

Al igual que los algoritmos anteriormente planteados, se partirá de una imagen a color que se transformará a escala de grises. De ésta forma, una variación de color se traducirá en una variación en la intensidad de gris. Puesto que se está trabajando con imágenes de 8 bits por pixel, el valor de cada pixel variará de 0 a 255, siendo el valor cero correspondiente a un color negro y el valor 255 a un blanco. Por tanto, cuanto más oscuro sea un color, más cercano estará su valor a cero.

Las imágenes que se van a analizar contienen uno o varios objetos que destacan sobre el fondo, siendo este último de un tono claro uniforme. Un ejemplo de imagen tomada se muestra en la figura 4.2.



Figura 4.2 Ejemplo de imagen tomada para el algoritmo de detección de borde.

Se decide tomar como criterio de discriminación la moda de la imagen en tonos grises. La imagen en escala de grises será una matriz llena de números que varían entre 0 y 255. Puesto que el fondo es uniforme y predominante, la mayoría de los píxeles tendrán el valor de la intensidad del fondo. La moda obtenida será, por tanto, prácticamente el valor del color de fondo, y servirá para diferenciar los objetos de él. Con sucesivas pruebas se comprueba que es necesario además añadirle un cierto umbral por las posibles sombras y variaciones de color.

Una vez obtenidos estos valores hay que analizar pixel a pixel la imagen. Se decide recorrerla de izquierda a derecha por filas. Cada vez que el puntero se encuentra en un pixel tiene que realizar dos comprobaciones. La primera es si el valor del pixel en el que está es menor que el umbral de discriminación y, además, la posición inmediatamente anterior es mayor. El umbral de discriminación no es más que la moda menos un cierto margen establecido de forma experimental. Si esta comprobación resulta positiva, el puntero se ha encontrado con el objeto, y por tanto con su contorno. La segunda sirve para saber en que momento se termina el objeto. Para ello se comprueba que el valor del pixel sea menor y que el inmediatamente posterior sea mayor. De esta forma, en la fila que se está analizando se ha detectado la entrada y la salida al objeto. Recorriendo cada fila, se encontrará el contorno del objeto. El diagrama de flujo que sigue esta función se muestra en la imagen 4.4.

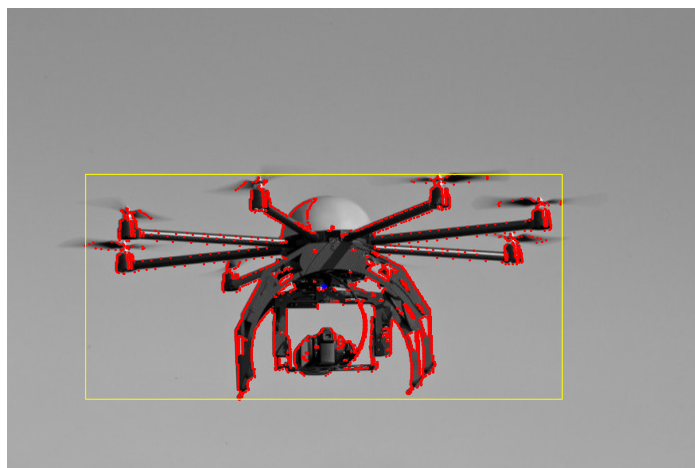


Figura 4.3 Resultado de la aplicación del algoritmo de detección de borde.

Como salida de esta función se obtienen dos vectores, X e Y. En ellos se han ido guardando las componentes j e i de los bordes detectados. Para visualizar el contorno del objeto basta con representar gráficamente estos vectores. El resultado se muestra en la imagen 4.3. En rojo se ha dibujado el contorno detectado; en azul, el centro aproximado del objeto; y en amarillo, una región rectangular que lo engloba.

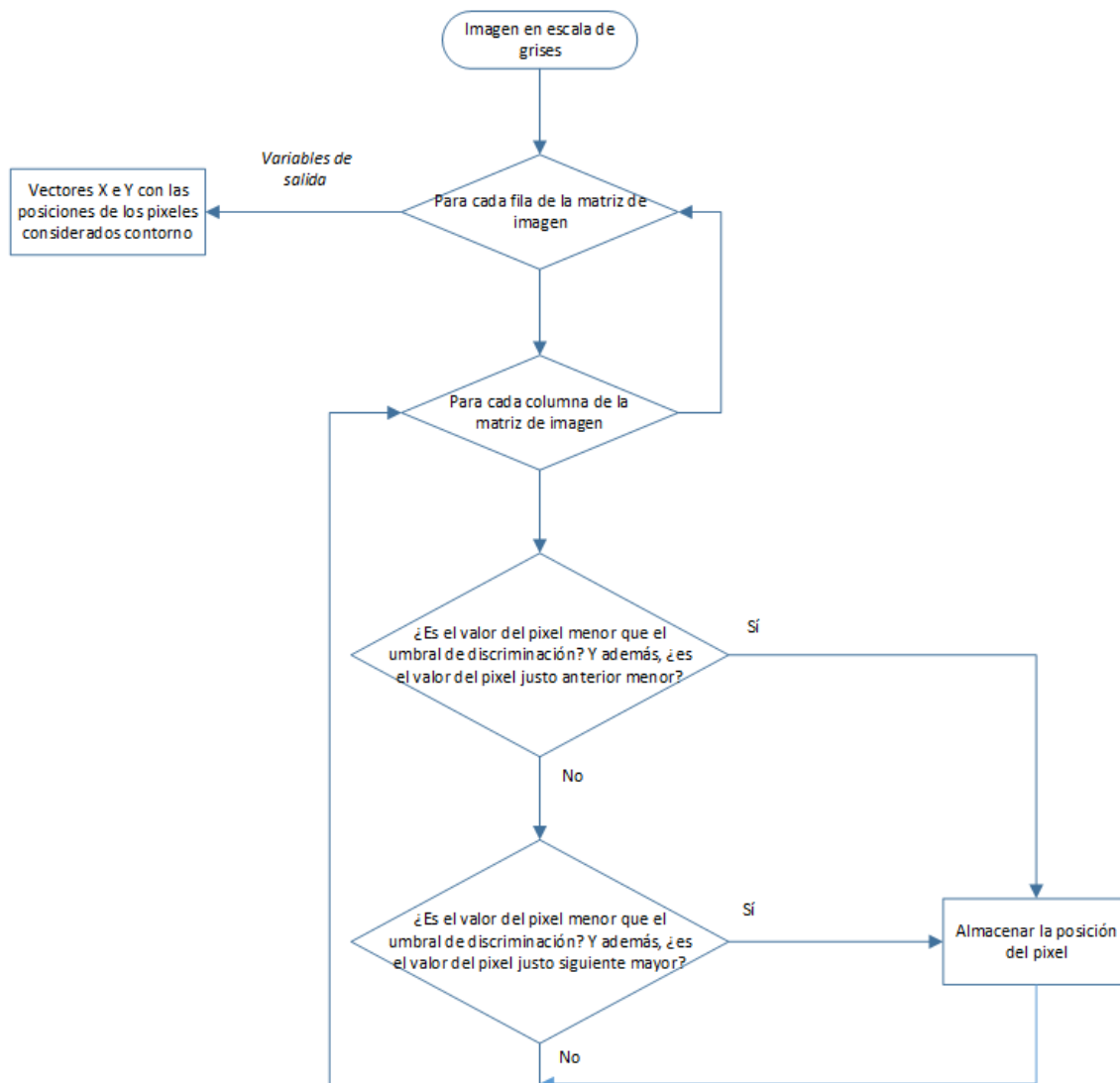


Figura 4.4 Algoritmo de detección de borde.

Incluyendo ciertas modificaciones en el código, se podría generalizar la detección para más de un objeto. Nuevamente con una imagen con fondo uniforme en el que destaquen los objetos, se obtiene la detección de los contornos de los objetos. En este caso solo se muestra en color azul el contorno de cada objeto detectado. El resultado se muestra en la imagen 4.5.



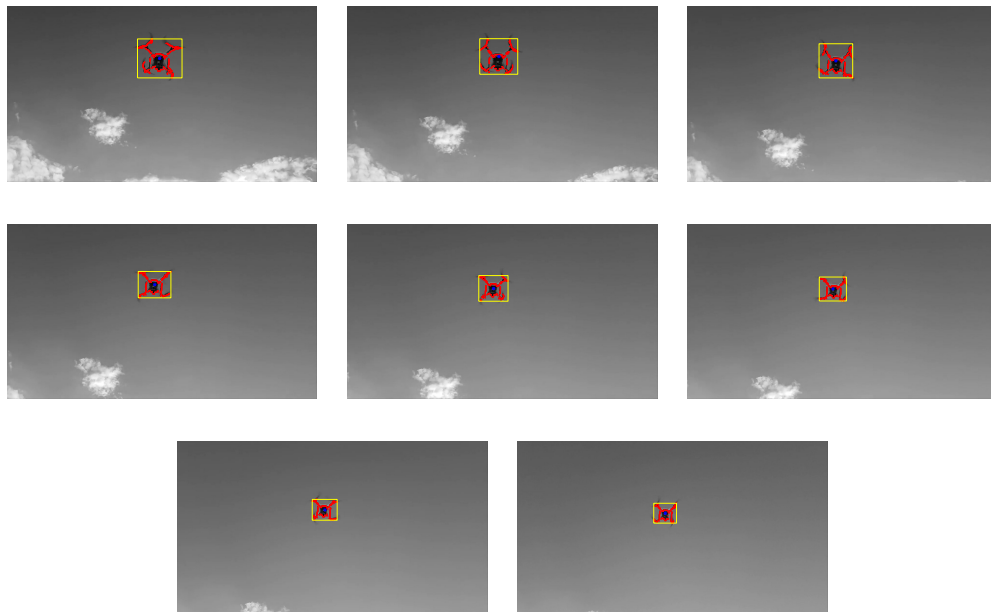
(a) Ejemplo de imagen que contiene más de un objeto.



(b) Aplicación del algoritmo a una imagen con más de un objeto.

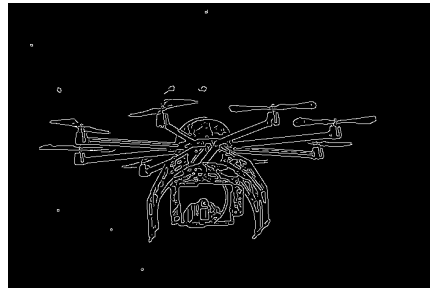
Figura 4.5 Ejemplo de detección de borde a más de un objeto.

Puesto que la idea desarrollada en este proyecto es ofrecer una alternativa a los sistemas anticolidión de una aeronave, con obtener el borde de un objeto en un instante determinado no sería suficiente. Por ello se plantea la aplicación del algoritmo a un fragmento de vídeo, de forma que se aplique progresivamente. En concreto, el vídeo recoge un vuelo de demostración de un *Phantom 4* (Fuente: [4]). Tras aplicar la detección el resultado obtenido se muestra en 4.6. Se han representado varios instantes del vídeo que ilustran el movimiento de dicho dron.

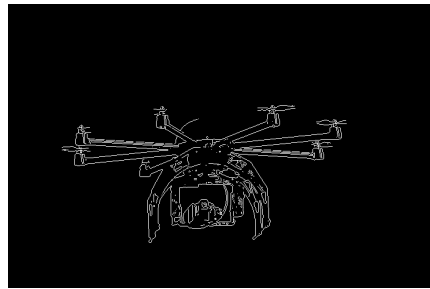
**Figura 4.6** Imágenes extraídas de la detección del contorno de un *Phantom 4* en movimiento.

Una vez presentados los resultados, se aplica ahora a las imágenes 4.2 y 4.5 los algoritmos de Canny y Sobel para comparar resultados. Como se puede observar, tanto para el caso de un sólo dron como para el de dos, el algoritmo de Canny reproduce el contorno al completo. Además es capaz de detectar también otras formas, como por ejemplo en la figura 4.8 se puede apreciar como ha detectado una nube presente en la

imagen a color.

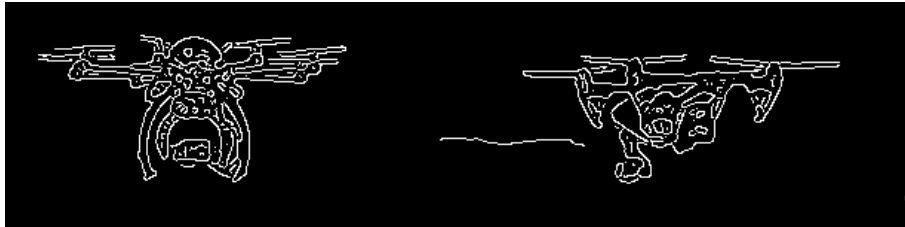


(a) Canny.

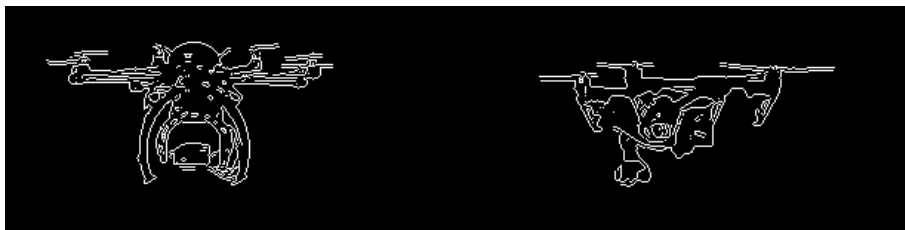


(b) Sobel.

Figura 4.7 Algoritmos de Canny y Sobel aplicados a la imagen 4.2.



(a) Canny.



(b) Sobel.

Figura 4.8 Algoritmos de Canny y Sobel aplicados a la imagen 4.5.

Frente a los resultados obtenidos, es evidente que los algoritmos ya creados e incluidos en *Matlab* son más eficientes en la detección de bordes. Por un lado, el algoritmo que se ha desarrollado no cumple todas las premisas necesarias para un detector de este tipo. Los contornos que devuelve no son cerrados y sólo es capaz de detectar el borde exterior. Además devuelve falsos negativos, puesto que hay contornos no detectados. Por otro lado, el algoritmo está basado únicamente en un descriptor de color. Si se añadiesen otras características, como el gradiente de la imagen, u otros descriptores, como de textura, forma o temperatura, se obtendrían

resultados más precisos.

Si bien es cierto que el resultado no es favorable, se trata de un resultado ya esperado. Tratar de competir con algoritmos de tal envergadura y aceptación en el mundo del procesamiento de imágenes era tarea casi imposible. Sin embargo, todo el trabajo realizado ha servido de cimientos para adentrarse en este área que tantísimas posibilidades ofrece.

4.2 Detección de objetos

El objetivo de esta sección es la detección de objetos en una imagen. En este caso, se trabaja con imágenes cuyo fondo es indiferente; ahora lo que importa es el color de los objetos. Como ya se explicó en capítulos anteriores, se están tratando imágenes en un espacio de color RGB. Éstas están formadas, en el ámbito digital, por tres matrices, una para cada color. Identificar objetos de colores rojo, verde o azul será trabajar con dichas matrices. El objetivo será encontrar en estas imágenes los colores que son de interés.

Para desarrollar esta detección se hará uso de un paquete de *Matlab*, el *Image Processing Toolbox*. Proporciona un amplio conjunto completo de algoritmos, funciones y aplicaciones, de referencia estándar para el análisis, visualización, procesado y tratamiento de imágenes. Además, la captura de imágenes se realiza con una aplicación que tiene integrada, llamada *imaqtool*. Ésta permite la adquisición de imágenes, la elección de su formato y su resolución, además de permitir tener varios dispositivos de captura de imágenes conectados a la vez.

Se parte por ejemplo de la imagen 4.9 en la que puede verse una lata de color rojo. El objetivo será por tanto detectar el color rojo dentro de esta imagen. Como se ha ido haciendo, el color se buscará en la imagen en escala de grises. El color rojo será detectado por un cambio de intensidad en el valor de los píxeles.

Se divide la imagen en las tres matrices correspondientes a los tres colores. Puesto que la de interés es la banda roja, se realiza una previsualización de la misma. Para aclarar la idea de las bandas de color, se han representado en la imagen 4.9 las tres bandas en paralelo. El color rojo se corresponde con las coordenadas (255, 0, 0). Se puede apreciar en la figura 2 de la imagen anterior que el objeto rojo ahora se visualiza en un tono gris claro. Si fuese un color rojo puro se vería completamente blanco, puesto que el valor 255 se corresponde con el blanco.

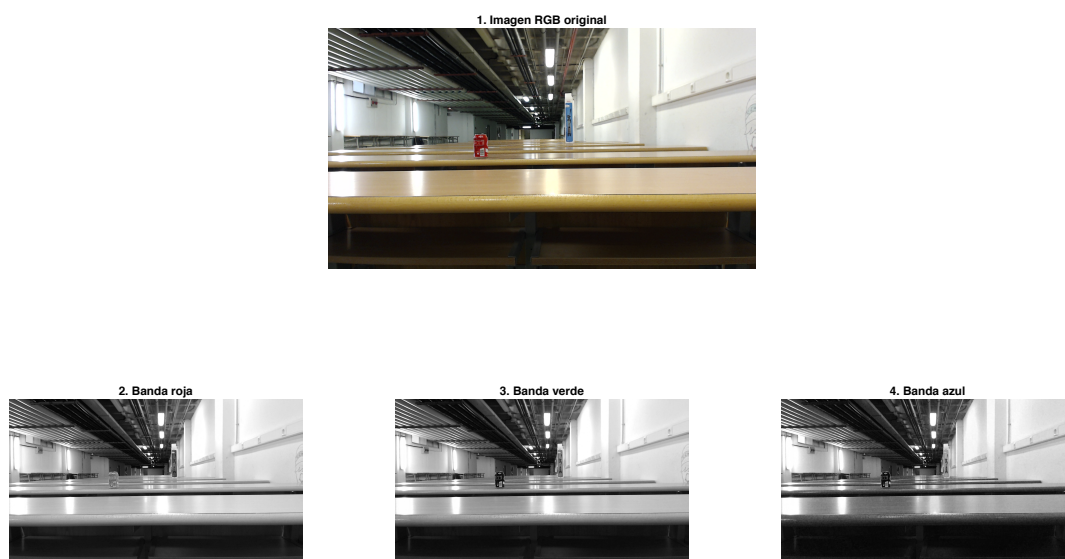


Figura 4.9 Visualización de las bandas de color de la imagen.

Ahora habrá que resaltar las zonas rojas de la imagen a nivel de gris. Como se está tratando matrices, modificar los colores en la imagen será tan simple como sumar y/o restarlas. Si se quiere destacar el rojo, bastará con eliminar los tonos azules y verdes de la imagen. Esta operación es tan simple como restar a la matriz de rojo las matrices de azul y verde. El resultado es una imagen en negro, donde se resaltan en tonos más claros los tonos rojos encontrados.

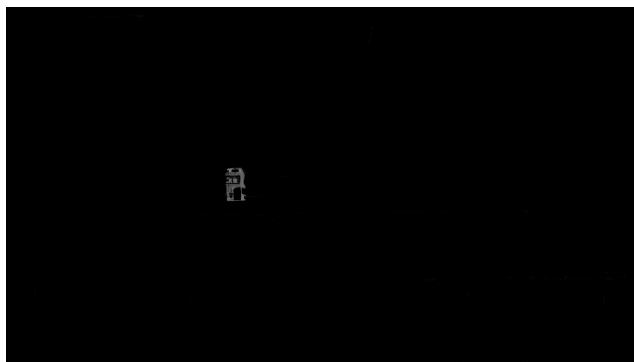


Figura 4.10 Zonas rojas resaltadas sobre la imagen.

Puesto que el fondo no tiene tonos puros ni uniformes, será necesario aplicar un filtro que elimine los posibles residuos. El filtro elegido es un filtro de mediana que ya incorpora *Matlab*, cuyo comando es `medfilt2`. Este filtro recorre la matriz por regiones, cuyo tamaño es modificable. Dentro de cada región, sustituye el valor del píxel estudiado por la mediana de los valores que engloba la región. El resultado obtenido se muestra en la imagen 4.11. En este caso quizá no es del todo apreciable la mejora obtenida, pero en otras ocasiones será indispensable para una buena detección.

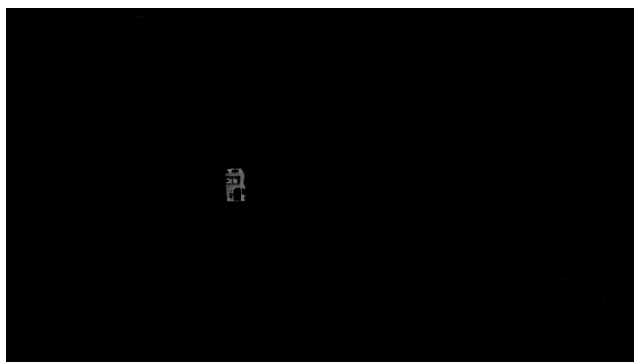


Figura 4.11 Imagen tras la aplicación del filtro de mediana.

Para resaltar más aún el color que es de interés, se transforma la imagen a binaria. Para ello se emplea la función `im2bw`. Como no se ha impuesto que los objetos sean de color uniforme, es posible que el objeto aparezca seccionado. Para solucionarlo, se aplica una nueva función, `imdilate`. Con ella, se recorre el contorno de cada sección con un elemento estructurante, y se agrandan los bordes de las figuras que haya en la imagen. Como elemento estructurante se ha elegido un disco de radio 15. Así, posibles objetos seccionados quedarán conexados. El resultado se muestra en la imagen 4.12.

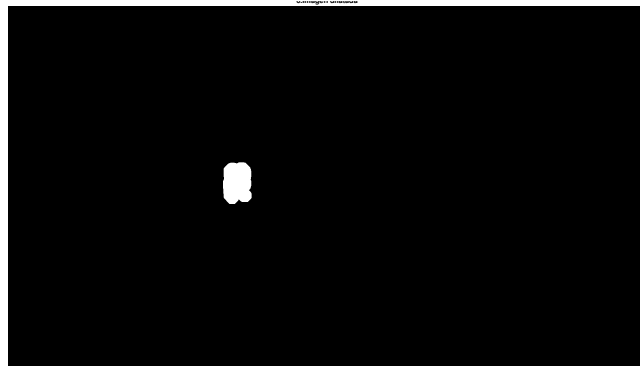


Figura 4.12 Imagen tras aplicar la dilatación a los bordes.

De manera similar, es posible extraer los colores verde y azul. En estos casos las bandas a resaltar serán la verde y la azul, respectivamente. Para ello habrá que restarle a la banda de interés las matrices de las otras dos. Los resultados obtenidos para el caso de objetos azules se muestran en la imagen 4.13.

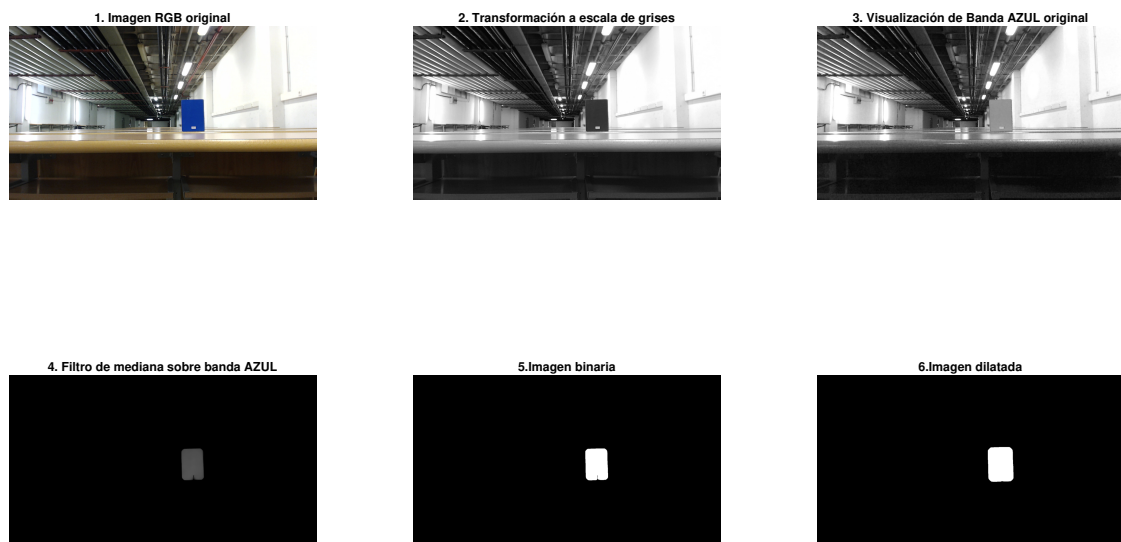


Figura 4.13 Detección de objeto azul.

Otra variación a tener en cuenta es la posibilidad de que haya más de un objeto. Sin tener que realizar ninguna modificación, todos los objetos son detectados correctamente. Por ejemplo, para el caso de objetos rojos, la detección se muestra en la imagen 4.14.

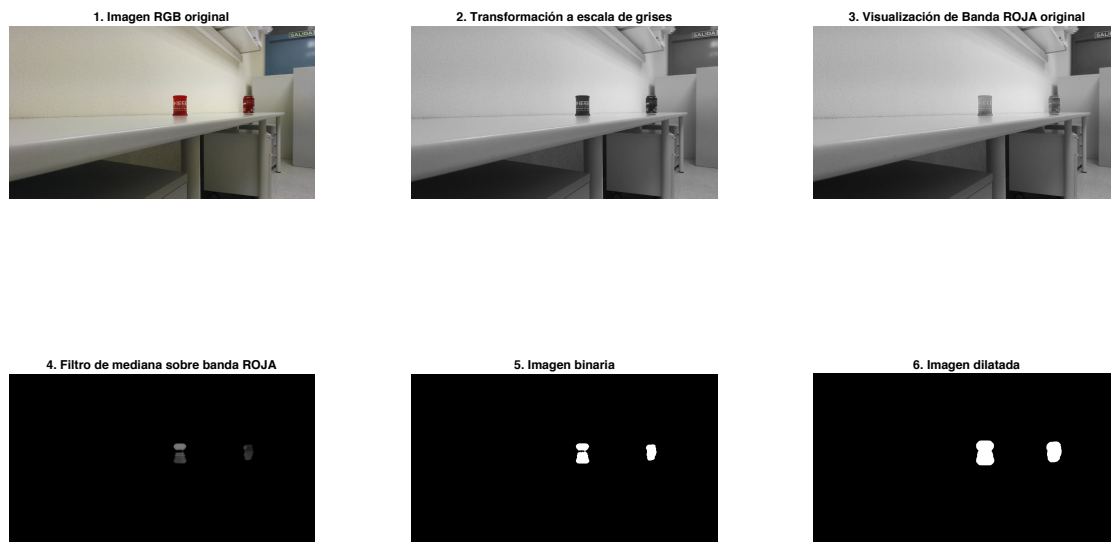


Figura 4.14 Detección de varios objetos simultáneamente.

En cuanto al algoritmo empleado se muestra un esquema en la figura 4.15. En él se han representado los pasos seguidos en la detección de objetos, desde la imagen de entrada RGB hasta la imagen de salida con la dilatación de los objetos. Entre los bloques se han indicado las funciones específicas de *Matlab* necesarias para llevar a cabo la operación.

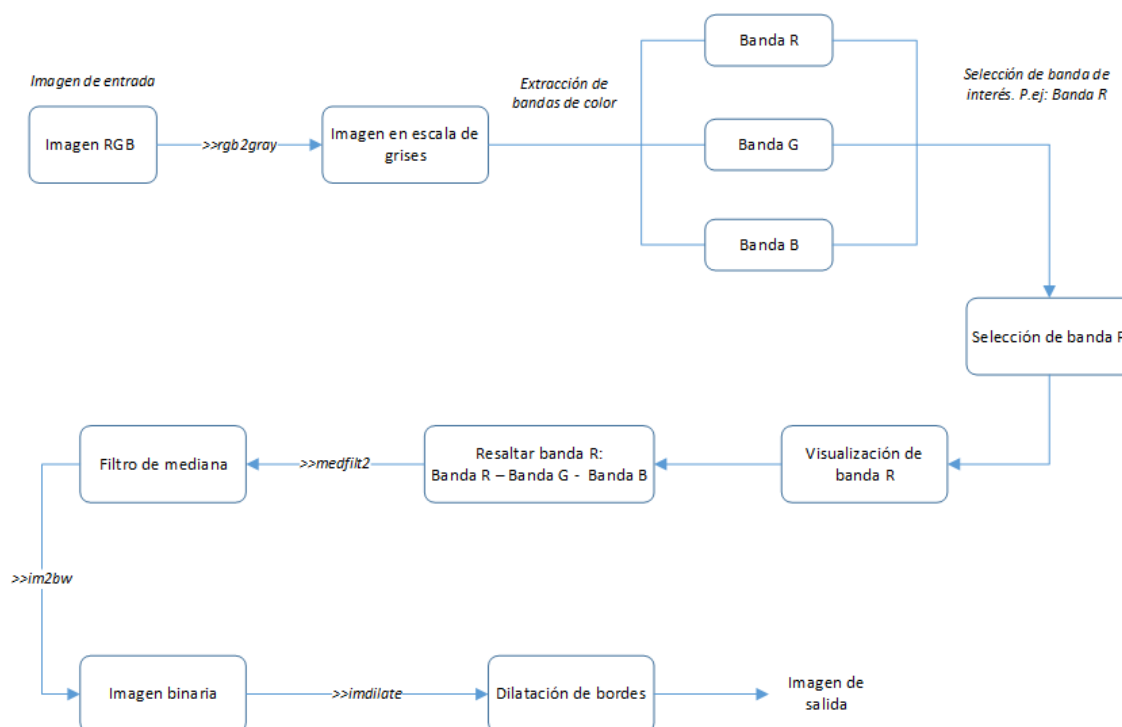


Figura 4.15 Esquema seguido en la detección de objetos.

Siguiendo la misma estructura que en el capítulo anterior, se prueba ahora el algoritmo de Canny sobre la imagen RGB de la figura 4.9. El objetivo es comparar la utilidad del algoritmo desarrollado frente al de Canny. Como se puede observar en la imagen 4.16, identificar el objeto de interés, en este caso la lata

roja, resulta muy complicado. De hecho, ni siquiera a simple vista se podría apreciar dónde se encuentra sin ver previamente la imagen RGB. Por ello, se puede concluir esta vez, que el algoritmo desarrollado para detección de objetos es totalmente válido: ofrece buenos resultados con un bajo coste computacional. Esta conclusión resulta de gran impacto para aplicaciones "indoor", por ejemplo en ámbitos industriales, donde se tratarían imágenes en entornos de este estilo.

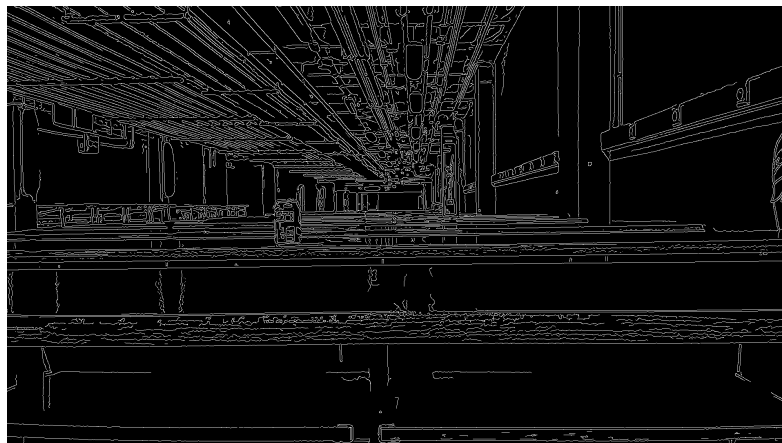


Figura 4.16 Aplicación de algoritmo de Canny a imagen 4.9.

4.3 Identificación de objetos

La siguiente sección está motivada por el concepto de visión estereoscópica que se usará más adelante. Con ella, se va a identificar el mismo objeto en dos imágenes distintas.

La visión estereoscópica artificial es una extensa área incluida dentro de la visión por computador. Es una de las herramientas de las que se dispone para conocer la estructura tridimensional de una imagen. Está basada en el procesamiento digital de dos imágenes diferentes, obtenidas mediante dos cámaras en distintas posiciones. La visión humana se basa también en la estereoscopia, donde los ojos ejercen la función de cámaras. Por tanto, la posición de las cámaras será similar a la de los ojos: alineadas y con una cierta separación entre ellas. Así, se obtendrán dos imágenes distintas, una izquierda y una derecha.

El punto de partida de esta sección es la detección de objetos desarrollada en la anterior. Se tienen ahora dos imágenes de la misma escena, tomadas desde dos posiciones diferentes. A ambas se le han aplicado todos los pasos anteriormente desarrollados. De esta forma se han obtenido dos imágenes binarias con un cierto número de objetos detectados, que no tiene por qué coincidir en ambas imágenes. Mediante la identificación de objetos se pretende asociar los objetos que han sido detectados en la imagen izquierda con los de la derecha. Para ello, se realizan dos comprobaciones diferentes.

La primera será comprobar que el tamaño del objeto coincida en ambas imágenes. Puesto que la forma del objeto puede variar de una imagen a otra, y con ellos su tamaño, se va a tomar un cierto umbral. Se trata de imágenes de resolución 1920x1080, luego se tienen un total de 2073600 píxeles. Se decide tomar un umbral entre 2000 y 3000, que constituyen aproximadamente un 0.1 % del total.

Para determinar el tamaño de cada objeto se parte de la imagen binaria dilatada anterior. El primer paso será realizar un *Labelling* o Etiquetaje. Con este paso se realiza una extracción de las componentes que se encuentren conexas. Para ello se emplea el comando `bwlabel()`. Dada una imagen binaria, devolverá una imagen con etiquetas numéricas, de forma que todos los píxeles que pertenecen a una misma región conectada compartirán la misma etiqueta. Habrá que tener en cuenta el tipo de conectividad que se quiere. En este caso se elige aplicar una conectividad a 8 por dar mejores resultados. Una vez se tienen las imágenes etiquetadas

hay que cuantificar el número de píxeles que tienen la misma etiqueta, y buscar un número parecido en la otra imagen. Puesto que el orden de etiquetado no tiene porque ser el mismo, habrá que comprobar todas las regiones conexas que se hayan encontrado.

La segunda comprobación a realizar guardará relación con la posición en la que se encuentra el objeto. Las imágenes que se obtienen de una cámara, al igual que las obtenidas por el ojo humano, tienen una cierta perspectiva cónica. Esta perspectiva nos ofrece una visión de la profundidad de forma parecida a como lo hace la estereoscopia. Siguiendo esta idea, el objetivo será encontrar el centro de cada objeto. Siempre que las dos cámaras estén alineadas, la altura del centro del objeto no cambiará. Para ello, se emplea el comando *»bwmorph*. Este comando realiza operaciones morfológicas a imágenes binarias. Una de las múltiples opciones que ofrece es la obtención del centro de los objetos detectados. Así, se obtendrá el centro de cada objeto detectado y se almacenará en dos vectores. Cuando en la comprobación anterior se detecte que dos objetos son el mismo, se comprobará que su altura sea la misma, permitiendo un cierto umbral por posibles fallos en la alineación de las cámaras.

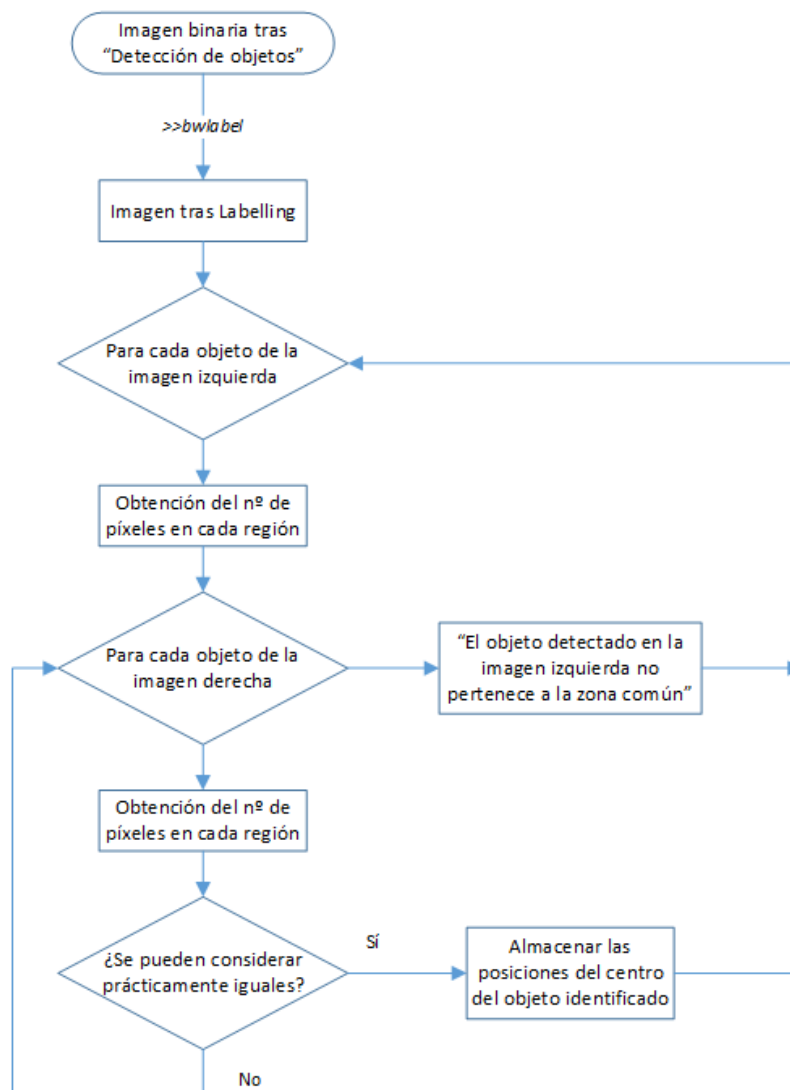


Figura 4.17 Primera comprobación en la identificación de objetos.

Tras la identificación, se obtiene una matriz con cuatro columnas y tantas filas como objetos haya. Dentro de la matriz, las columnas uno y dos almacenan las componentes x e y del centro de los objetos de la imagen izquierda. Las columnas tres y cuatro almacenan los centros de los objetos de la imagen derecha. Cada fila se

corresponderá, por tanto, con un objeto.

La identificación de objetos se ha implementado teniendo también en cuenta el color de ellos. Primero se han detectado de forma independiente los objetos de colores rojo, verde y azul. Así, se han obtenido las correspondientes imágenes binarias y los centros de los objetos. Posteriormente se ha ejecutado la identificación de objetos para cada bloque de color. Como resultado se tienen tres matrices como la explicada anteriormente, una para cada color.

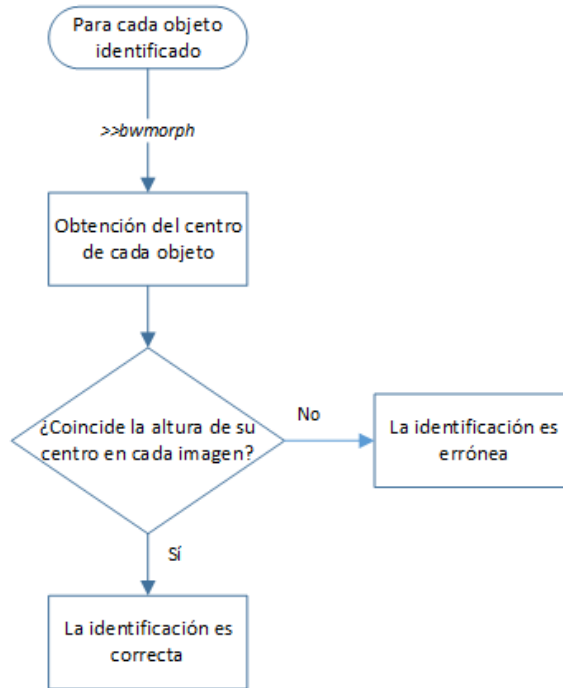


Figura 4.18 Segunda comprobación en la identificación de objetos.

4.4 Cálculo de distancia

El objetivo último del procesamiento de imágenes que se ha realizado es el cálculo de distancia a objetos. Métodos de obtención de la distancia hay muchos, pero pueden ser clasificados en activos y pasivos. Un método de medición de distancia activo es aquel que envía señales a los objetos, como láser, ultra sonido, señales de radio, etc. Si el método es pasivo, le basta con recibir información de la posición de los objetos. Como ya se anticipó anteriormente, la medición estereoscópica es un ejemplo de método pasivo. Para su realización se necesita hacer uso de dos cámaras, con dos restricciones. Por un lado, las cámaras deberán estar alineadas horizontalmente. Por otro, las imágenes deben ser tomadas en el mismo instante de tiempo. Tratándose de situaciones estáticas, esta última restricción no es relevante.

Para el desarrollo del presente proyecto, se parte de un montaje físico con dos cámaras Logitech c920 HD PRO WebCam. Sus especificaciones técnicas pueden verse en la página web del fabricante. Las cámaras se colocan sobre una barra de aluminio de un metro de longitud. Puesto que incorporan una pinza de sujeción, se podrán colocar a la distancia que se desee, sin necesidad de realizar perforaciones en el aluminio. La barra se encuentra fijada a un trípode de altura variable. Además, se ha fijado un nivel de burbuja a ella, para asegurar la alineación horizontal de las cámaras. El montaje realizado puede verse en la imagen 4.19.

Para la obtención de la distancia se ha seguido el esquema de la figura 4.20. En él, la imagen izquierda se toma desde la posición S_L y la derecha desde S_R . El ángulo φ_0 es el ángulo de apertura horizontal de la cámara y B representa la distancia entre las cámaras. Para obtener la distancia D hasta el objeto sólo hay que

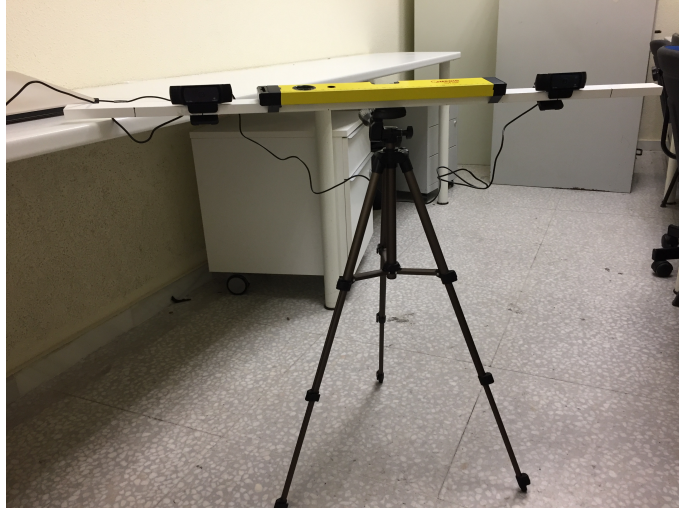


Figura 4.19 Montaje físico realizado.

usar geometría.

Por un lado, si los ejes de la cámara son paralelos, la distancia entre las cámaras se puede expresar como indica 4.3. B_1 y B_2 representan las distancias horizontales de las cámaras izquierda y derecha, respectivamente, al objeto. ϕ_1 y ϕ_2 son los ángulos formados entre el eje de cada cámara y el objeto. Estas variables están representadas en la imagen 4.21.

$$B = B_1 + B_2 = D \tan \phi_1 + D \tan \phi_2 \quad (4.3)$$

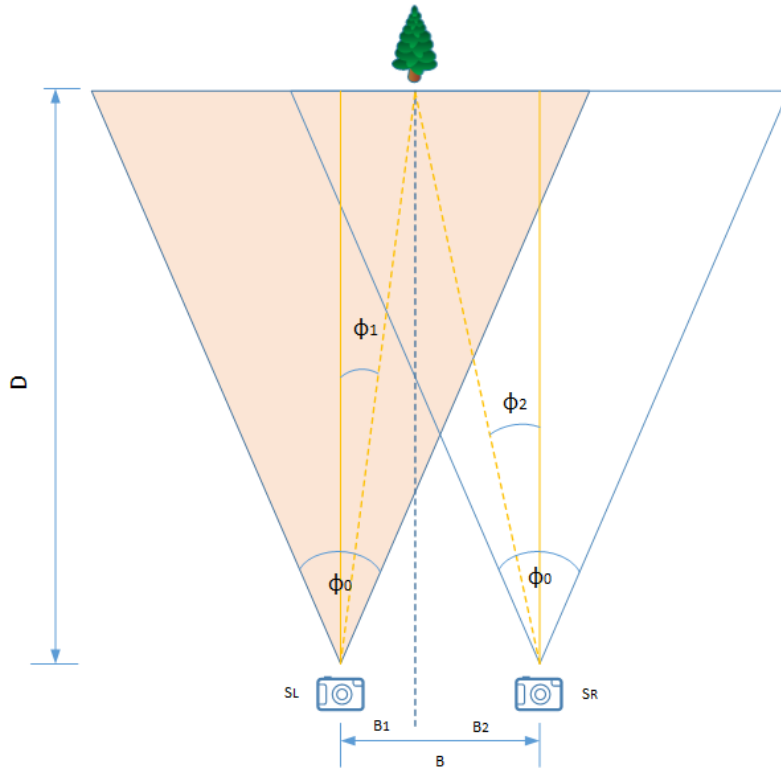


Figura 4.20 Esquema geométrico seguido en el cálculo de la distancia.

Despejando la distancia D de la ecuación anterior, se obtiene la expresión 4.4.

$$D = \frac{B}{\tan\phi_1 + \tan\phi_2} \quad (4.4)$$

Mediante semejanza de triángulos, de la imagen 4.21 se obtienen las expresiones 4.5 y 4.6. Las variables x_1 y x_2 son las distancias del eje de cada cámara al objeto en píxeles y $\frac{x_0}{2}$ es la mitad de la resolución horizontal de la cámara, también en píxeles.

$$\frac{x_1}{\frac{x_0}{2}} = \frac{\tan\phi_1}{\tan\left(\frac{\phi_0}{2}\right)} \quad (4.5)$$

$$\frac{-x_2}{\frac{x_0}{2}} = \frac{\tan\phi_2}{\tan\left(\frac{\phi_0}{2}\right)} \quad (4.6)$$

Si se despeja de las ecuaciones anteriores $\tan\phi_1$ y $\tan\phi_2$, y se suman los resultados, se tiene la expresión 4.7.

$$\tan\phi_1 + \tan\phi_2 = 2\tan\left(\frac{\phi_0}{2}\right) \left(\frac{x_1}{x_0} - \frac{x_2}{x_0}\right) \quad (4.7)$$

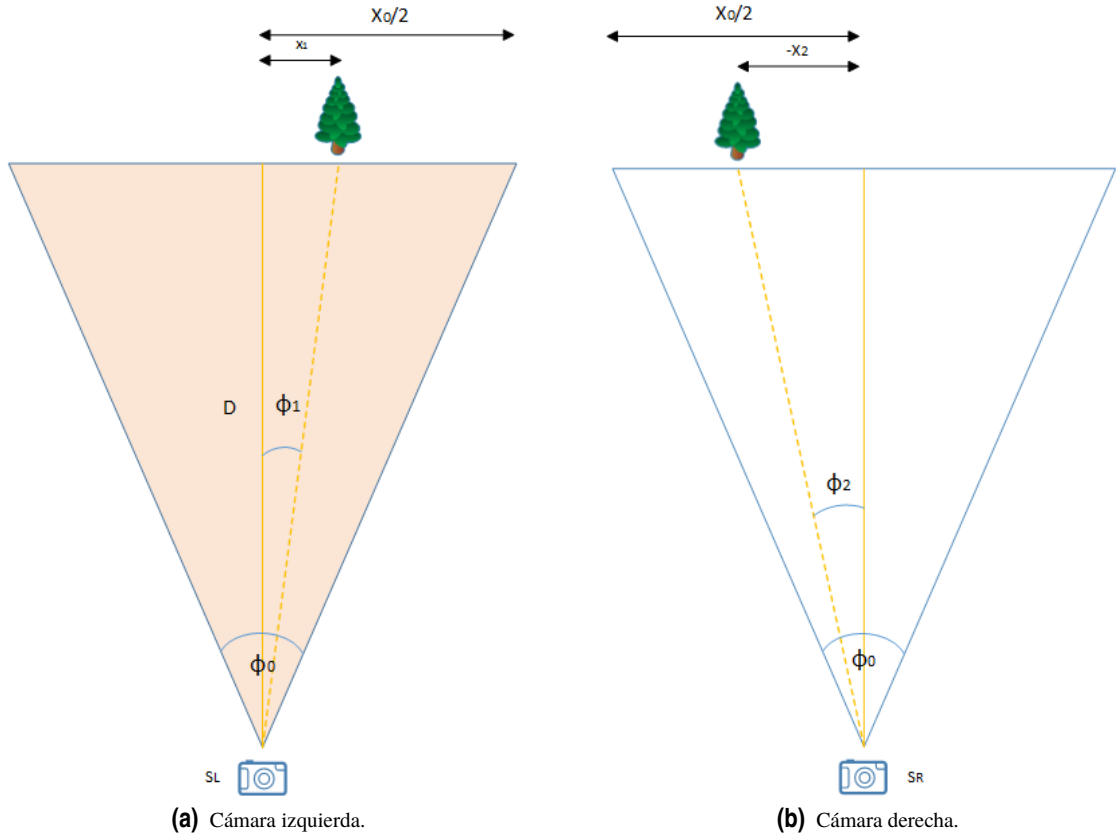


Figura 4.21 Esquema geométrico de cada cámara.

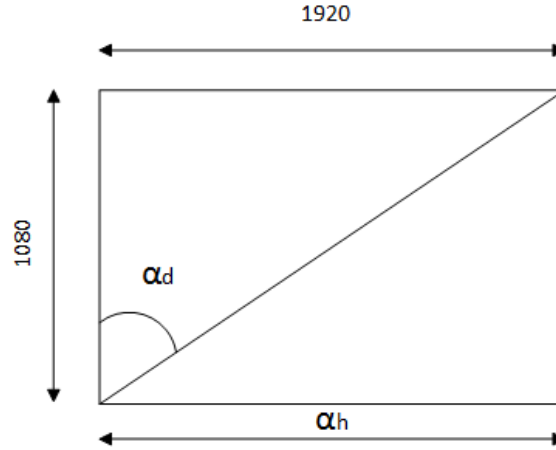


Figura 4.22 Ángulo de visión horizontal.

Sustituyendo esta última en la ecuación 4.4 se puede expresar la distancia como en 4.8.

$$D = \frac{Bx_0}{2 \tan\left(\frac{\phi_0}{2}\right) (x_1 - x_2)} \quad (4.8)$$

Del resultado anterior, se deduce que conociendo la distancia entre las cámaras (B), la resolución horizontal de las cámaras (x_0), el ángulo de apertura horizontal ϕ_0 y la diferencia ($x_1 - x_2$), se puede conocer la distancia al objeto. [16]

La distancia B será un dato conocido, puesto que la posición de las cámaras será fijada de antemano. El montaje realizado permite colocar las cámaras a cualquier distancia entre máximo de 80 cm y un mínimo de 20 cm.

La resolución también será conocida y elegida. Para la adquisición de las imágenes se va a emplear la herramienta lde *Matlab*, »*imaqtool*. Como se explicó anteriormente, este comando ejecuta una nueva pantalla que permite elegir con qué cámara se quiere realizar la adquisición de la imagen, con qué resolución y en qué formato, entre otras muchas opciones. De esta forma, la variable x_0 será conocida.

En cuanto al ángulo de apertura horizontal, se trata de un dato que depende única y exclusivamente de la cámara. En la ficha técnica del producto facilitada en la página web del fabricante se encuentra el dato de ángulo de apertura diagonal, concretamente 78° . Conociendo la resolución vertical y horizontal de la imagen y la apertura diagonal, se puede obtener el dato que hace falta con un sencillo cálculo. De las ecuaciones 4.9 se obtiene un valor de α_h de 67.98° , ángulo que se puede verse representado en la imagen 4.22. Este valor será recalculado durante el proceso de obtención de distancia, buscando una convergencia del mismo.

$$\tan \alpha = \frac{1920}{1080} \Rightarrow \sin \alpha = \frac{\alpha_h}{\alpha_d} \quad (4.9)$$

Para obtener la diferencia ($x_1 - x_2$) es necesario determinar a qué objeto se quiere calcular la distancia. Puesto que en las secciones anteriores se ha llegado a la identificación de los objetos a partir de su centro, se escogerá este punto como representación del objeto. Para obtener el valor de x_1 basta con restar a la coordenada x del centro del objeto en la imagen izquierda, la mitad de la resolución horizontal, como se puede intuir de la imagen 4.21. Para x_2 , se procede de la misma forma, pero con el centro obtenido en la imagen derecha.

Una vez conocidas todas las variables necesarias para el cálculo de la distancia, se debe desarrollar un algoritmo que lo implemente. Para ello, bastará con ensamblar cada uno de los bloques que se han ido

analizando progresivamente e incluir estas últimas consideraciones. El objetivo es conseguir el cálculo de la distancia a objetos de colores rojo, verde y azul, pues son los que han sido detectados anteriormente. El esquema seguido se muestra en la imagen 4.23.

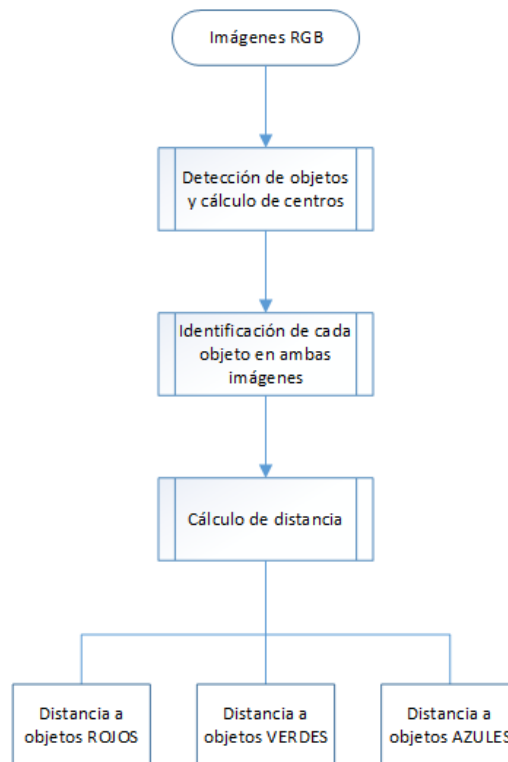


Figura 4.23 Algoritmo de cálculo de distancia.

4.5 Resultados

En esta sección se presentan los resultados obtenidos tras la aplicación del algoritmo de cálculo de distancia. Se han desarrollado una variedad de ensayos en diferentes situaciones, con el fin de extraer información acerca de la precisión que ofrece.

En el primer ensayo se propone el estudio de objetos azules que se encuentran centrados entre las cámaras. Éstas se han fijado con una distancia entre ellas de 80 cm. Se prueban tres distancias arbitrarias: 112 cm, 126 cm y 158 cm. El valor del ángulo de visión se va recalculando mediante iteraciones. Partiendo del ángulo inicialmente calculado se calcula la primera distancia. Conociendo la distancia real se recalcula el ángulo de la misma ecuación 4.8. Con sucesivas iteraciones se obtiene una media del ángulo de visión horizontal de 70.38° .

En la imagen 4.24 se puede ver un ejemplo del tipo de imagen empleada durante el primer ensayo.

Tabla 4.1 Tabla obtenida del Ensayo 1.

	$B(\text{cm})$	$\alpha_h(^{\circ})$	Altitud en imagen izq (pixel)	Altitud en imagen dcha (pixel)	D. real (cm)	D. medida (cm)
Prueba 1	80	73.07	482	492	112	107.1
Prueba 2	80	67.98	455	459	126	132.7
Prueba 3	80	70.78	491	507	158	164.8



Figura 4.24 Ejemplo de imágenes para el ensayo 1.

En vista a los resultados obtenidos en la tabla 4.1, se calculan los errores relativos para las tres pruebas. En el primer caso se obtiene un error del 2 %, en el segundo de 5.3 % y finalmente en el tercero de 4.3 %. En este caso, puesto que se han empleado diferentes valores de ángulo de visión horizontal, los errores no siguen ninguna relación con el aumento o disminución de la distancia.

Se realiza ahora un nuevo ensayo, igual que el anterior, pero en este caso los objetos detectados son de color rojo. El objetivo es comprobar que la detección de ambos colores se realiza correctamente, y que no influye en el cálculo de la distancia. Entre las cámaras se mantiene una distancia de 80 cm. En este caso, el ángulo se fija al valor medio obtenido anteriormente. Las distancias siguen siendo arbitrarias, pero se intercalan con las elegidas anteriormente, con el fin de tener más variedad en los resultados. Concretamente el objeto se sitúa a 107 cm, 132 y 165. En la imagen 4.25, se puede observar un ejemplo de las imágenes tomadas para el ensayo.



Figura 4.25 Ejemplo de imágenes para el ensayo 2.

En la tabla de resultados 4.2, se tienen unos errores relativos de 2.7 %, 4.6 % y de 5.4 %. En este caso, con valores fijos de separación entre cámaras y de ángulo de visión, se tiene una tendencia creciente en el error relativo durante el cálculo de distancia, de forma que cuanto mayor sea la distancia a la que se encuentra el objeto, mayor es el error. Observando la posición del centro del objeto, se aprecian ciertas diferencias entre las imágenes tomadas con la cámara izquierda y con la cámara derecha. Estas variaciones se deben a pequeños errores en la alineación de las cámaras. Esto induce a pensar que en el caso de que la alineación entre las cámaras fuese muy precisa, los errores relativos se reducirían considerablemente.

Tabla 4.2 Tabla obtenida del Ensayo 2.

	$B(\text{cm})$	$\alpha_h(^{\circ})$	Altitud en imagen izq (pixel)	Altitud en imagen dcha (pixel)	D. real (cm)	D. medida (cm)
Prueba 1	80	70.38	491	483	107	109.9
Prueba 2	80	70.38	509	514	132	138.04
Prueba 3	80	70.38	517	519	165	173.98

En el siguiente ensayo se quiere comprobar si el cálculo de la distancia es igual de preciso cuando los objetos no se encuentran centrados entre las cámaras. En el desarrollo planteado en la sección anterior, la distancia B se ha dividido en dos tramos independientes, B_1 y B_2 . Estos tramos se corresponden a las distancias horizontales entre cada cámara y el objeto. Por ello, a priori los resultados obtenidos deberían tener un precisión comparable a la obtenida en el ensayo anteriormente realizado.

Se sitúan ahora los objetos rojos desviados del centro de las cámaras, a ambos lados. Las distancias nuevamente son intercaladas entre las anteriores. Un ejemplo de imagen con el objeto desviado a la derecha, se muestra en la figura 4.26.



Figura 4.26 Ejemplo de imágenes para el ensayo 3.

Los resultados obtenidos se muestran en la tabla 4.3. Los errores cometidos para los tres casos son de 4.3 %, 2.7 % y 5.3 %. Nuevamente los resultados obtenidos presentan un pequeño error relativo. Analizando los datos obtenidos, se puede concluir que los objetos no tienen porqué estar centrados entre las cámaras. Como ya se había anticipado, la distancia B introducida en la ecuación de cálculo de distancia es independiente de dónde esté situado el objeto. Además, para la obtención de la ecuación 4.8, se han tratado de manera independiente las cámaras izquierda y derecha.

Tabla 4.3 Tabla obtenida del Ensayo 3.

	$B(cm)$	$\alpha_i(^{\circ})$	Altitud en imagen izq (pixel)	Altitud en imagen dcha (pixel)	D. real (cm)	D. medida (cm)
Prueba 1	80	70.38	490	506	114	118.9
Prueba 2	80	70.38	503	516	124	127.4
Prueba 3	80	70.38	518	518	198	208.7

En el siguiente análisis se pretende comprobar que no solamente sirve para imágenes con un objeto y que no necesariamente todos los objetos tienen que ser del mismo color. Se plantea una situación como la mostrada en la imagen 4.27. En ella pueden verse dos objetos, uno de color rojo y otro de color azul, colocados a distintas distancias de las cámaras. De nuevo, los resultados mostrados en la tabla 4.4 tienen un error relativo de 4.3 % para el objeto rojo y 3.9 % para el azul.



Figura 4.27 Ejemplo de imágenes para el ensayo 4.

Tabla 4.4 Tabla obtenida del Ensayo 4.

	$B(cm)$	$\alpha_h(^{\circ})$	Altitud en imagen izq (pixel)	Altitud en imagen dcha (pixel)	D. real (cm)	D. medida (cm)
Objeto Rojo	80	70.38	510	507	144	150.2
Objeto Azul	80	70.38	506	495	109	113.3

En este caso se tienen tres objetos, dos de ellos azules y uno rojo. El objetivo es el mismo que el anterior, comprobar que sigue ofreciendo los mismos resultados cuando hay más de un objeto, y cuando hay más de un objeto de un color. En la imagen 4.28, el objeto azul 1 que se muestra en la tabla se corresponde con el más pequeño, mientras que el objeto 2 se corresponde con el más grande.



Figura 4.28 Ejemplo de imágenes para el ensayo 5.

Se obtienen unos errores relativos de 3.5 %, 6.1 % y 3.5 %. En este caso llama la atención que el objeto azul que está más alejado presenta un error menor que el que se encuentra más cercano. Esto es debido a que el objeto 2 es mayor que el objeto 1 y además es de color uniforme. Así, la detección es más precisa que para un objeto más pequeño.

Tabla 4.5 Tabla obtenida del Ensayo 5.

	$B(cm)$	$\alpha_h(^{\circ})$	Altitud en imagen izq (pixel)	Altitud en imagen dcha (pixel)	D. real (cm)	D. medida (cm)
Objeto Rojo	80	70.38	503	503	135	139.8
Objeto Azul 1	80	70.38	443	494	110	116.7
Objeto Azul 2	80	70.38	474	482	170	175.9

Una vez analizada la influencia de la distancia de los objetos a las cámaras y su posición respecto al centro, se propone el análisis de la influencia de la distancia entre las cámaras. Para ello se parte de la imagen 4.28,

y se varía la distancia entre las cámaras entre 80cm, 60cm, 35cm y 20cm. En la tabla 4.6 se muestran los resultados organizados para cada objeto. Estos revelan que cuanto más cerca se encuentren las cámaras entre sí, mayor es el error relativo. También muestra que a mayor distancia del objeto a las cámaras, mayor es el error cometido.

Tabla 4.6 Tabla obtenida del ensayo 6.

Objeto 1: Lata roja					
$B(cm)$	$\alpha_h(^{\circ})$	Altitud en imagen izq (pixel)	Altitud en imagen dcha (pixel)	D. real (cm)	D. medida (cm)
80	70.38	503	503	135	139.8
60	70.38	502	504	135	142.6
35	70.38	502	525	135	234.7
20	70.38	510	531	135	344.7
Objeto 2: Lata azul					
$B(cm)$	$\alpha_h(^{\circ})$	Altitud en imagen izq (pixel)	Altitud en imagen dcha (pixel)	D. real (cm)	D. medida (cm)
80	70.38	443	494	110	116.7
60	70.38	447	499	110	121.9
35	70.38	454	526	110	167.2
20	70.38	501	526	110	226.9
Objeto 3: Libreta azul					
$B(cm)$	$\alpha_h(^{\circ})$	Altitud en imagen izq (pixel)	Altitud en imagen dcha (pixel)	D. real (cm)	D. medida (cm)
80	70.38	474	482	170	175.9
60	70.38	477	481	170	181.5
35	70.38	477	497	170	350.3
20	70.38	480	503	170	633.2

Los resultados obtenidos en la serie de ensayos realizados revelan varias conclusiones. Por un lado, si la distancia a la que se encuentra el objeto es mayor, el resultado obtenido presenta un error mayor. El hecho de que el objeto se encuentre más lejos, implica a nivel digital que su tamaño en pixel es menor. Esto quiere decir que, un error en la detección del centro de un objeto de 5 pixeles, por ejemplo, afecta mucho más a un objeto lejano que a uno cercano. Es decir, los objetos alejados presentan un menor margen de error en la alineación de las cámaras.

Por otro lado, también puede observarse que todas las distancias medidas superan en valor las distancias reales, salvo una. Esta medida es del ensayo 1, cuando se estaba calibrando el ángulo de visión horizontal. La distancia real es de 112 cm y la distancia medida de 107.1 cm. En ese caso concreto se emplea un ángulo de 73.07, valor superior al establecido posteriormente. Si se analiza ese mismo caso con el ángulo de 70.38, se obtiene un valor de 113.6, que efectivamente se encuentra por encima.

La tabla 4.6, revela que a mayor distancia entre las cámaras, mejor es la precisión. Los errores se disparan cuando la distancia se reduce más de 60 grados. Cuanto más cercas estén entre sí, más parecidas serán las imágenes obtenidas y por tanto, menor será la diferencia en la posición del objeto en cada imagen. Esta diferencia se encuentra en el denominador de la ecuación 4.8, $(x_1 - x_2)$. Por tanto es evidente que cuanto menor sea esta diferencia mayor será el valor de distancia calculada. Se puede concluir que para obtener unos buenos resultados habrá que mantener las cámaras a una distancia de unos 80 cm.

En definitiva, los resultados obtenidos con cámaras suficientemente separadas muestran un error pequeño, probablemente reducible si se optimizan todas las variables, como alineación, calibración, medidas de distancia reales, etc.

5 Conclusiones

En este capítulo se exponen las conclusiones obtenidas tras la realización del proyecto.

Desde un punto de vista cuantitativo, analizando los resultados obtenidos en los diferentes ensayos y pruebas realizadas, se tiene de media un error de entre el 4 y el 5 %. Se trata de un error aceptable, pero mejorable, como posteriormente se detallará en el siguiente capítulo. No se ha tenido en cuenta los resultados obtenidos en casos en los que las cámaras se encontraban excesivamente juntas. Como ya se demostró anteriormente, cuando la distancia entre las cámaras es pequeña, el error presenta una rápida tendencia creciente. Esto es así porque el denominador del cociente que proporciona la distancia (ecuación 4.8) tiende a cero más rápidamente que el numerador. Este último detalle es de vital importancia a la hora de analizar sobre qué tipo de plataformas se podría implementar. En pequeños UAVs, habría que considerar reemplazar las cámaras por otras cuya lente disponga de unas características que le permitan situarse más cerca.

Por tanto, se ha desarrollado un algoritmo de cálculo de distancia sencillo y de bajo coste computacional, pero con una buena proyección de futuro por la facilidad que ofrece para complementarlo con información adicional.

En cuanto al estudio de mejora en los sistemas de navegación, se trata de un primer paso en cuanto a sistemas anticolidión. La visión computacional constituye una herramienta extremadamente potente para el posicionamiento de objetos. Si bien es obvio que el sistema TCAS actual ofrece una buena resolución de conflictos, aún presenta carencias en cuantos a vehículos no tripulados. Pese al esfuerzo por el desarrollo del futuro ACAS X, se trata de un mercado en el que, como en todos, la competencia produce avances en la tecnología. También es importante desarrollar sistemas asequibles a todos los sectores. Por ejemplo, los UAVs en el sector del ocio, no pueden hacer frente a inversiones tan grandes como para equiparlos con sistemas anticolidión de tal envergadura. Ni siquiera algunos podrían hacer frente al peso que supondría.

Puesto que no se trata de una propuesta de sistema anticolidión completo, también podría verse no como competencia, si no como complemento perfecto para la redundancia de información y un aumento en la seguridad aérea. Se podría lograr una reducción de las distancias mínimas de seguridad, reduciendo así el volumen de seguridad. Esto conllevaría a largo plazo, una reducción de las pérdidas que le suponen tanto a clientes como compañías aéreas la congestión del espacio. También es de interés el impacto que generaría en cuanto al ATM. A mayor autonomía por parte de las aeronaves, mayor capacidad tendrá el ATM. De esta forma, se podrá proporcionar servicios de Navegación Aérea a un mayor volumen de tráfico aéreo. Se lograría una mayor optimización del sistema, y por tanto una mayor eficiencia. Como ha quedado reflejado, se trata de un tema de interés europeo que debe ser solucionado de cara a un futuro no muy lejano.

En conclusión, desde un punto de vista cualitativo, es un primer paso para adentrarse en una nueva propuesta de mejora para la navegación aérea. Incluir la visión computacional podría proporcionar la información necesaria que logre descongestionar el espacio aéreo y aumentar los niveles de seguridad.

6 Líneas futuras

En este capítulo se presentan las posibles líneas de trabajo futuras, así como mejoras encontradas en el proyecto.

Desde un punto de vista particular, en el algoritmo de cálculo de distancia se deben tener en cuenta varias consideraciones. Por un lado, las cámaras necesitan de una calibración previa con el fin de lograr la perfecta alineación horizontal necesaria para obtener el mínimo error en los cálculos. Con vista al objetivo último de formar parte de un sistema de navegación, lo óptimo sería realizar una calibración continua y automática, para poder hacer frente a las vibraciones a las que se encuentre sometido el montaje. Sin embargo, la calibración no ha sido objeto de este proyecto. Por ello, se trata de una posible línea futura en cuanto a mejora de los resultados.

Otro punto a destacar es la determinación de la zona común entre las cámaras. Cada cámara de forma independiente tiene su propio campo de visión, en el que es capaz de detectar una serie de objetos. Cuando estos campos de visión se ponen en común, no todos los objetos son detectados por ambas cámaras. Por ello, se debería iniciar una criba inicial previa a la identificación de objetos, en la que se distinguiesen qué objetos son detectados por ambas cámaras y cuáles no.

Desde un punto más general, centrándose en la navegación aérea se plantean también posibles líneas de trabajo futuras. Puesto que los sistemas de navegación no se emplean en situaciones estáticas, se debería considerar el cálculo de distancia a objetos en movimiento. En la práctica esto no sería más que realizar una conversión de vídeo a imágenes, con el tiempo de captura de imágenes que se desee. De esta forma se ha procedido en el apartado de detección de bordes, en la imagen 4.6. Con el cálculo de distancia realizado a varias imágenes, y el tiempo entre imágenes, podría calcularse también la velocidad a la que se mueve el objeto.

Todas las propuestas anteriores incluyen mejoras para complementar este trabajo. Pero sin duda, donde más trayectoria futura se puede encontrar es en el campo del procesamiento de imágenes. Se trata de un campo extremadamente extenso donde se ofrecen múltiples posibilidades. Por ejemplo, en este caso se ha tratado de utilizar como descriptor de imagen el color. Si se usan descriptores de forma, temperatura (con cámaras infrarrojas), textura, etc. se podría obtener cualquier objeto en cualquier entorno, que es el cometido último de la mejora al sistema de navegación que se plantea.

Además, es importante resaltar la versatilidad que ofrece la extracción de información de imágenes. Se trata de una aplicación no sólo válida para ámbitos aeronáuticos, si no también automovilísticos e incluso medio ambientales. Por ejemplo, para la detección de manchas de contaminación en el océano, la detección de objetos a través de imágenes podría servir para determinar su tamaño, profundidad, movimiento, etc.

En definitiva, se ha iniciado un camino con mucho trayecto aún por recorrer, que puede ofrecer una innovación a los sistemas de navegación, con la consecuente descongestión del espacio aéreo.

Índice de Figuras

2.1	En la imagen a) el sistema BCAS (1970); en la b), el TCAS (1980-2000). <i>Fuente: [13]</i>	4
2.2	Accidente aéreo en San Diego, en 1978. <i>Fuente: [13]</i>	4
2.3	Tipos de interfaz para TCAS. <i>Fuente: [11]</i>	6
2.4	Simbología empleada por TCAS. <i>Fuente: [11]</i>	7
2.5	Volumen de protección en vista vertical. <i>Fuente: [11]</i>	8
2.6	Volumen de protección en vista horizontal. <i>Fuente: [11]</i>	9
2.7	Situaciones de peligro generadas por la malinterpretación del aviso "Adjust Vertical Speed". <i>Fuente: [2]</i>	10
2.8	Comparación de las alertas sonoras "Adjust Vertical Speed, Adjust" y "Level off, Level off". <i>Fuente: [2]</i>	11
2.9	Mejora incluida en la versión 7.1, donde la aeronave equipada, evita la colisión gracias a un nuevo aviso RA. <i>Fuente: [2]</i>	11
2.10	Funcionamiento interno de ACAS X. <i>Fuente: [13]</i>	12
3.1	Espacio de color RGB. <i>Fuente: [5]</i>	16
3.2	Imagen RGB compuesta por tres matrices de la misma dimensión	16
3.3	Comparación entre detector local, b), y detector global, c)	18
3.4	Aplicación del descriptor DCD con 8 colores dominantes	19
3.5	Imágenes diferentes cuyo histograma es el mismo	20
3.6	Proceso seguido por el descriptor CLD	21
3.7	Formas posibles del borde en una imagen. <i>Fuente: [5]</i>	23
3.8	Borde del tipo T	23
3.9	Ejemplo de seguimiento de contorno	25
4.1	Imagen comparativa de detección de borde	29
4.2	Ejemplo de imagen tomada para el algoritmo de detección de borde	30
4.3	Resultado de la aplicación del algoritmo de detección de borde	30
4.4	Algoritmo de detección de borde	31
4.5	Ejemplo de detección de borde a más de un objeto	32
4.6	Imágenes extraídas de la detección del contorno de un <i>Phantom 4</i> en movimiento	32
4.7	Algoritmos de Canny y Sobel aplicados a la imagen 4.2	33
4.8	Algoritmos de Canny y Sobel aplicados a la imagen 4.5	33
4.9	Visualización de las bandas de color de la imagen	34
4.10	Zonas rojas resaltadas sobre la imagen	35
4.11	Imagen tras la aplicación del filtro de mediana	35
4.12	Imagen tras aplicar la dilatación a los bordes	36
4.13	Detección de objeto azul	36
4.14	Detección de varios objetos simultáneamente	37
4.15	Esquema seguido en la detección de objetos	37
4.16	Aplicación de algoritmo de Canny a imagen 4.9	38
4.17	Primera comprobación en la identificación de objetos	39
4.18	Segunda comprobación en la identificación de objetos	40

4.19	Montaje físico realizado	41
4.20	Esquema geométrico seguido en el cálculo de la distancia	41
4.21	Esquema geométrico de cada cámara	42
4.22	Ángulo de visión horizontal	43
4.23	Algoritmo de cálculo de distancia	44
4.24	Ejemplo de imágenes para el ensayo 1	45
4.25	Ejemplo de imágenes para el ensayo 2	45
4.26	Ejemplo de imágenes para el ensayo 3	46
4.27	Ejemplo de imágenes para el ensayo 4	47
4.28	Ejemplo de imágenes para el ensayo 5	47

Índice de Tablas

4.1	Tabla obtenida del Ensayo 1	44
4.2	Tabla obtenida del Ensayo 2	45
4.3	Tabla obtenida del Ensayo 3	46
4.4	Tabla obtenida del Ensayo 4	47
4.5	Tabla obtenida del Ensayo 5	47
4.6	Tabla obtenida del ensayo 6	48

Bibliografía

- [1] F. Barfield, *Autonomous collision avoidance: the technical requirements*, Proceedings of the IEEE 2000 National Aerospace and Electronics Conference. (2000).
- [2] Airborne Collision and Avoidance Systems, *ACAS Guide*, (2016).
- [3] The European Commission, (2011).
- [4] DJI, *Página oficial del fabricante de phantom 4*, en youtube.
- [5] Nicolás Luis Fernández, *Contribución al reconocimiento de objetos 2d mediante detección de bordes en imágenes a color*, 2002.
- [6] Óscar Boullosa García, *Estudio comparativo de descriptores visuales para la detección de escenas cuasi-duplicadas*, (2011).
- [7] M. Prats I. Finazzi, V.Fico, *New algorithm for horizon detection and attitude estimation*, (2015).
- [8] Icao, *Aeronautical Telecommunications*, Volume IV Surveillance Radar and Collision Avoidance Systems **IV** (2002).
- [9] ICAO, *Airborne Collision Avoidance System (ACAS) Manual*, (2006).
- [10] International Civil Aviation Organization, *Rules of the Air*, Annex 2 to the Convention on International Civil Aviation (2012).
- [11] IVAO HQ training department, *Airborne Collision Avoidance System - TCAS*, (2015).
- [12] Yazdi Ibrahim Jenie, *Autonomous collision avoidance system for UAVs*, PhD Proposal (2013).
- [13] Mykel J Kochenderfer, Jessica E Holland, and James P Chryssanthacopoulos, *Next-Generation Airborne Collision Avoidance System*, Lincoln Laboratory Journal **19** (2013).
- [14] B S Manjunath, Jens-rainer Ohm, and Vinod V Vasudevan, *Color and texture descriptors*, **11** (2001).
- [15] Thiago Toshio Matsumoto, Lucio Flavio Vismari, Ricardo Alexandre Veiga Gimenes, Jorge Rady de Almeida, and Joao B. Camargo, *A Learning-Based Autonomous Control System Approach for Collision Avoidance within an Unmanned Aircraft*, 2015 IEEE International Conference on Dependable Systems and Networks Workshops (2015).

- [16] Jernej Mrovlje and Damir Vran, *Distance measuring based on stereoscopic pictures*, 9th International PhD Workshop on Systems and Control: Young Generation Viewpoint **2** (2008).
- [17] J. Valverde Rebaza, *Detección de bordes mediante el algoritmo de canny*, (2007).
- [18] C. Transportes S. Kallas, *Cielo único: la comisión adopta medidas para poner fin a la congestión del espacio aéreo europeo*, (2013).
- [19] Robert M. Sabin, Michael J. y Gray, *Global convergence and empirical consistency of the generalized lloyd algorithm*, (1986).
- [20] Hong Shao and Xu Wang, *Chest X-ray retrieval based on modified Homogeneous Texture Descriptor and local feature fusion*, 2012 5th International Congress on Image and Signal Processing, CISP 2012 (2012).
- [21] International Standards, Recommended Practices, T O The, Convention On, International Civil, Part Ii, International Civil Aviation, and International Commercial, *Operation of Aircraft*, 2010.
- [22] Split Tracks, *NET ALERT - the Safety Nets newsletter ACAS X – the future of airborne collision avoidance*, (2013).
- [23] A. Valenzuela, *Gestión del tráfico aéreo*, (2015).
- [24] Carlos Ventura-Royo, *Image-Based Query by Example Using MPEG-7 Visual Descriptors*, (2010).
- [25] Ka Man Wong, Lai Man Po, and Kwok Wai Cheung, *Dominant color structure descriptor for image retrieval*, Proceedings - International Conference on Image Processing, ICIP **6** (2006).

Glosario

Algoritmo de Lloyd Generalizado También llamado algoritmo K-means, se trata de un algoritmo de método de agrupamiento, que tiene como objetivo la partición de un conjunto de n observaciones en k grupos en el que cada observación pertenece al grupo cuyo valor medio es más cercano. Utiliza una técnica de refinamiento iterativo. 19

Funciones de Gabor Se trata de funciones que están presentes tanto en el dominio espacial como en el de la frecuencia. Es un paquete de ondas gaussiano, es decir, con una envolvente gaussiana que la localiza espacialmente. Fueron propuestas como un método óptimo de descomponer una señal en cuantos de información. 21

RVSM Término usado en aviación para la separación vertical mínima requerida entre dos aeronaves. Mediante la introducción de este concepto se redujo la separación vertical de 2000 ft a 1000ft para niveles de vuelo por encima de FL290. Con ello se consiguió aumentar los niveles de vuelo disponibles. Además, las aeronaves pueden volar más cerca de su nivel de vuelo óptimo, lo que supone un ahorro de combustible. 9

Transformada de Coseno Discreta Expresa una secuencia finita de varios puntos como resultado de la suma de distintas señales sinusoidales (con distintas frecuencias y amplitudes). Trabaja con una serie de números finitos y sólo con cosenos. Es una función lineal e invertible del dominio real R_N al dominio real R_N . 20, 21

Siglas

ACAS Airborne Collision Avoidance System. VII, IX, 1, 3–14

ADS-B Automatic Dependent Surveillance-Broadcast. 7, 12

ATC Air Traffic Control. 3, 10, 11

ATCRBS Air Traffic Control Radar Beacon System. 4

ATM Air Traffic Management. 12

BCAS Beacon Collision Avoidance System. 3, 4

CLD Color Layout Descriptor. IX, 18, 20, 21

CSD Color Structure Descriptor. IX, 18, 20

CSD Contour-based Shape Descriptor. 22

DCD Dominant Color Descriptor. IX, 18, 20

EHD Edge Histogram Descriptor. 21

EUROCONTROL European Organization for the Safety of Air Navigation. 4

FAA Federal Aviation Administration. 4, 12

GA General Aviation. 12

GoF Group of Frame. 18

HTD Homogeneous Texture Descriptor. 21, 22

ICAO International Civil Aviation Organization. 4, 5, 7, 9, 11, 13

MCTOM Maximum Certified Take-Off Mass. 11

MTOW Maximum Take-Off Weight. 8

NextGen Next Generation Air Transportation System. 12, 13

PA Proximity Advisories. 8

RA Resolution Advisories. 5, 7–12

RF Radio Frequency. 5

RGB Red Green Blue. 15, 19

RSD Region-based Shape Descriptor. 22

SCD Scalable Color Descriptor. 18

SES Single European Sky. 1

SESAR Single European Sky Research. 12, 13

SNR Signal-to-Noise Ratio. 24

SSR Secondary Surveillance Radar. 3, 8

SVO Selective Velocity Obstacle. 13

TA Traffic Advisories. 5, 7–9

TBD Texture Browsing Descriptor. 21

TCAS Traffic Alert and Collision Avoidance System. VII, IX, 3–5, 8, 9, 11, 12, 49

UAS Unmanned Aerial System. 13

UAV Unmanned Aerial Vehicle. IX, 1, 12, 13, 49

UCAV Unmanned Combat Aerial System. 13

VO Velocity Obstacle. 13